

# NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA



19960220 042

## THESIS

### **AUTOMATED WATCHBILL GENERATION AT NAVY COMMANDS: A PROTOTYPE SYSTEM FOR FNMOC**

by

Virginia T. Lamb and Theresa C. Michal

September, 1995

Thesis Co-Advisors: Hemant K. Bhargava  
Roger Stemp

**Approved for public release; distribution is unlimited.**

THIS QUALITY INSPECTED 1

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE Sep 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis		
4. TITLE AND SUBTITLE AUTOMATED WATCHBILL GENERATION AT NAVY COMMANDS: A PROTOTYPE SYSTEM FOR FNMOC		5. FUNDING NUMBERS		
6. AUTHOR(S) Virginia T. Lamb and Theresa C. Michal				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words)  This thesis investigates the feasibility of providing an easy to use automated system for developing watchbills that can be used by commands throughout the United States Navy. Currently the generation of watchbills is a manual process which is both time consuming and frustrating. This thesis develops a limited model of a working prototype of a fully automated system to be tested by Fleet Numeric Meteorological Oceanographic Command. A requirements study was conducted to determine the needs for the new system, data models designed to implement the requirements, and a feasible automated system developed.  The system is implemented in Microsoft Access using a series of clearly designed and visually pleasing forms to step the user through the watchbill generating process in a well defined and efficient manner.				
14. SUBJECT TERMS Automated Watchbill Generator			15. NUMBER OF PAGES 129	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18 298-102



Approved for public release; distribution is unlimited.

**AUTOMATED WATCHBILL GENERATION AT NAVY COMMANDS: A  
PROTOTYPE SYSTEM FOR FNMOC**

by

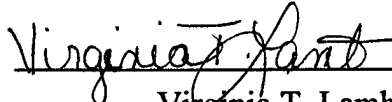
Virginia T. Lamb  
Lieutenant , United States Navy  
B.S., Jacksonville University, 1990  
Theresa C. Michal  
Lieutenant, United States Navy  
B.S., United States Naval Academy, 1988

Submitted in partial fulfillment  
of the requirements for the degree of

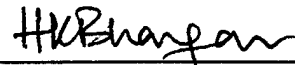
**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY  
MANAGEMENT**

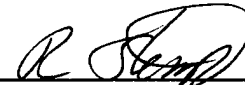
from the  
**NAVAL POSTGRADUATE SCHOOL**  
September 1995


Authors:

  
Virginia T. Lamb and Theresa C. Michal

Approved by:

  
Hemant K. Bhargava, Co-Advisor

  
Roger Stemp, Co-Advisor

  
Reuben Harris, Chairman  
Department of Systems Management



## **ABSTRACT**

This thesis investigates the feasibility of providing an easy to use automated system for developing watchbills that can be used by commands throughout the United States Navy. Currently the generation of watchbills is a manual process which is both time consuming and frustrating. This thesis develops a limited model of a working prototype of a fully automated system to be tested by Fleet Numeric Meteorological Oceanographic Command. A requirements study was conducted to determine the needs for the new system, data models designed to implement the requirements, and a feasible automated system developed.

The system is implemented in Microsoft Access using a series of clearly designed and visually pleasing forms to step the user through the watchbill generating process in a well defined and efficient manner.



## TABLE OF CONTENTS

I. INTRODUCTION .....	1
A. BACKGROUND .....	1
B. OBJECTIVES .....	1
C. SCOPE .....	2
D. METHODOLOGY .....	3
E. ORGANIZATION OF STUDY .....	3
II. SCHEDULING SYSTEMS .....	5
A. SCHEDULING DIFFICULTIES .....	5
B. MANUAL SCHEDULING .....	5
C. FLEXIBILITY .....	6
D. PRIORITY CONCEPTS .....	7
E. SCHEDULE EFFICIENCY .....	8
F. COMPARING ALTERNATIVE SCHEDULING SYSTEMS .....	8
1. Schedule Optimization .....	8
2. Priorities .....	9
3. Timing .....	9
4. Stability and Changes .....	9
5. Effect of the System for the User .....	9
G. CENTRALIZATION VS DECENTRALIZATION .....	10
H. DEGREE OF AUTOMATION FOR THE SCHEDULE .....	10
III. REQUIREMENTS ANALYSIS .....	11
A. WATCH ORGANIZATION .....	11
B. FNMOC REQUIREMENTS STUDY .....	14
1. Watch Positions and Rotations .....	14
2. FNMOC Personnel .....	15



C.	FNMOC CURRENT SCHEDULING SYSTEM .....	16
D.	FNMOC SYSTEM REQUIREMENTS .....	16
IV.	METHODOLOGY .....	19
A.	SEMANTIC OBJECT MODELING .....	19
1.	Attributes .....	19
2.	Instances .....	20
B.	CONCEPTUAL DESIGN .....	21
1.	Personnel Object .....	21
2.	WatchBill Object .....	22
3.	Snivel Object .....	23
4.	Semantic Object Model Reports .....	23
C.	DATA CONSIDERATIONS .....	23
1.	Personnel Database .....	24
2.	Input Requirements .....	24
3.	Output Reports .....	24
V.	IMPLEMENTATION .....	27
A.	DBMS SOFTWARE SELECTION .....	27
B.	MICROSOFT ACCESS OVERVIEW .....	28
1.	Database Tables .....	28
2.	Queries .....	29
3.	Database Forms .....	29
4.	Macros .....	30
5.	Modules .....	30
6.	Reports .....	31
C.	IMPLEMENTATION OF FNMOC WATCH BILL GENERATOR ...	31
1.	Table Creation .....	31
a.	Personnel Table .....	32

b. Watch Bill Table .....	33
c. Snivel Table .....	34
2. System Queries .....	35
3. System Modules .....	36
a. AssignWatchStanders .....	36
b. PreAssignWatchstanders .....	37
c. AssignValueWithUserInput .....	37
d. DeleteWatchstanders .....	37
4. System Forms .....	38
a. Personnel Forms .....	38
b. Snivel Forms .....	39
c. Generate Watchbill Forms .....	39
5. View Watchbill .....	40
6. Reports .....	40
 VI. CONCLUSION AND RECOMMENDATIONS .....	 43
A. LESSONS LEARNED .....	43
B. PROBLEMS ENCOUNTERED .....	44
C. CONCLUSIONS .....	45
D. RECOMMENDATIONS .....	45
 APPENDIX A. SEMANTIC OBJECT REPORTS .....	 49
 APPENDIX B. DATABASE TABLES .....	 67
 APPENDIX C. ACCESS DATABASE CODE .....	 71
 APPENDIX D. USER'S MANUAL .....	 97

LIST OF REFERENCES .....	117
--------------------------	-----

INITIAL DISTRIBUTION LIST .....	119
---------------------------------	-----

## **I. INTRODUCTION**

### **A. BACKGROUND**

Organizations have plans of operation for personnel scheduling. This plan may be a very elaborate one outlined in standard operating procedures or it may be left to the experience and intuition of the scheduler (Poshyanonda, 1989). The U.S. Navy schedules personnel into duty sections using schedules referred to as watchbills. According to the Navy Standard Operations Regulations Manual, a watch is defined as

any period of time during which an individual is assigned specific, detailed responsibilities on a recurring basis. The Commanding Officer will establish the watches necessary for the safety, security, and proper operation of the command.

The current process of watch bill scheduling involves a combination of standard operating procedures, experience, and intuition of the scheduler. The steps involved in watch bill generation are performed manually in a similar fashion throughout the Navy, thus the process readily lends itself automation. Automating the process will eliminate the extensive man-hours required to generate and update the watch bill thus allowing the coordinator the ability to focus on his/her primary duties. Any automated system must be robust enough in its design to account for all the factors if it is to be truly useful to the scheduler (Poshyanonda, 1989).

### **B. OBJECTIVES**

The purpose of this thesis is to develop a limited automated watchbill generator prototype to be used by Fleet Numerical Meteorological Oceanography Center (FNMOC), Monterey, California. The model is designed to integrate the scheduling logic of FNMOC's

Senior Watch Officer (SWO) into the prototype. The resulting system is expected to reduce the extensive man-hours required to generate and update the schedule in order to produce an efficient watchbill.

### **C. SCOPE**

A complete scheduling system provides several major functions, including capability for data extraction, storage and retrieval, and models and algorithms for generating optimal schedules. Additionally, the system maintains a user interface that offers several representations for problem data and solutions, and mechanisms for managing user interactions. The scope of this thesis includes the following tasks:

1. Conduct a review of the Navy's watchbill generating regulations.
2. Gather data from individuals having experience as Senior Watch Officers in order to determine requirements and expectations of a watchbill system.
3. Write appropriate modules and macros to access databases, extract, download, and load relevant data.
4. Write applications to analyze data, generate schedules, and display solutions.

The deliverable is :

1. A list of the requirements for an automated watchbill program capable of being implemented at FNMOC.
2. Working prototype using Microsoft Access.
3. A user's manual for the proper operation of the system.

## **D. METHODOLOGY**

The thesis follows a structured methodology for database development, which consists of the following steps (Elmasri/Navathe, 1989):

1. **Requirements Study:** This step includes interviewing the users and gathering data indicating the present state of the scheduling procedures at FNMOC.
2. **Conceptual Design:** The use of the Semantic Object data modeling to represent the data requirements.
3. **Logical Design:** Transforming the conceptual design into a logical and usable design.
4. **Implementation:** Using a commercial database management system, the logical design is transformed into a physical database.

## **E. ORGANIZATION OF STUDY**

This thesis is organized as follows: Chapter II describes scheduling systems and manual processes. Chapter III is the requirements study of FNMOC current system and identifies areas for automation. Chapter IV discusses methodology of design transformation. Chapter V describes implementation procedures and software requirements. Chapter VI discusses a lessons learned and benefits of the study.



## **II. SCHEDULING SYSTEMS**

This chapter discusses a general definition and view of scheduling as it would be in any type of management organization. Scheduling concepts and principles addressed include scheduling difficulties, manual scheduling procedures, flexibility, priorities, efficiency, schedule alternative comparison justification, and degree of system automation determination.

### **A. SCHEDULING DIFFICULTIES**

Effective scheduling is a knowledge intensive activity requiring a comprehensive model of the organization and its environment at all times (O'Brien, 1969). There are two main issues of complexity and uncertainty that make scheduling solutions difficult. The first difficulty is due to the size of the search space and the process of finding a sequence of actions that will satisfy scheduling objectives that change based on domain specific information. The complexity of the scheduling environment increases the number of possible combinations to be searched. The second difficulty arises due to hierarchial solution perspectives used in the generation of schedules for constraining the search space. There is a need for an intelligent scheduling system that can coordinate the relationships among interacting sub-problems and goals (Poshyanonda, 1989). A systematic planning and scheduling system would ease complicated tasks which require a large amount of expertise, good coordination among functions and accurate data (Hartman and Howard, 1976).

### **B. MANUAL SCHEDULING**

The task of scheduling is complex and there are a variety of alternatives for each of the choices the scheduler has to make. Many things can disrupt the schedule such as when



a scheduler overlooks one of the factors, an unexpected emergency arises, or a piece of necessary equipment breaks down (Hartman and Howard, 1976). The scheduler must be flexible and the people executing the schedule must have the initiative to re-arrange their work and develop an alternate plan (Poshyanonda, 1989). Additionally, the scheduler may be faced with short deadlines, and personnel changes made at the last moment (Bow, 1966)

With a manual scheduling process, a scheduler can only consider a few of these alternatives. There is more information in the system than the manual scheduler can handle. The speed and information processing capacity of the computer suggest that an automated scheduling system can consider more information and more alternatives, and may be able to efficiently produce error free schedules, yet Navy watchbill scheduling in the 1990's remains largely a manual process (Hartman and Howard, 1977).

### **C. FLEXIBILITY**

One aspect of the scheduling process is the dynamic nature of the operating environment. A system that can react intelligently to changing circumstances is necessary (Poshyanonda, 1989), thus in order for a scheduling system to be useful it must be flexible to the environment. The system should be able to function with minimum input, yet be able to handle additional data. A complete solution to the scheduling problem should include the function of predictive planning coupled with an ability to alter schedules in response to changing conditions (Poshyanonda, 1989). It should be able to use information based on priorities, but also produce schedules even if priorities are not assigned (O'Brien, 1969). The schedule should reflect personnel preferences of working hours and days. In times of high demand, schedules need to be produced in spite of expressed preferences. The system must

provide for manual override of the rare situation which it cannot handle routinely (Hartman and Howard, 1976). An automated system provides added flexibility and allows for more alternatives and more choices of options. In any circumstances, the system always produces a recommended schedule.

#### **D. PRIORITY CONCEPTS**

In any effective scheduling system, there must be some means for dealing with priority relationships among the various personnel (Hartman and Howard, 1976). The extent to which priorities are established and used is a decision of the scheduler, but the system should be capable of handling whatever priorities are assigned. At one extreme, the system may be designed in which no priorities are assigned and all personnel are treated equally with no allowance for preferences, while the other extreme every person may be allowed a specific amount of points which measures their ranking in a priority hierarchy (Hartman and Howard, 1977). The major issue in discussing priorities is not the assignment of priority numbers, but rather the establishment of clear, well defined, and agreed upon rules which explain what preferences rank in priority over one another (Poshyanonda, 1989).

Adherence to priority rankings will restrict the possible choices open to the scheduler whether scheduling is done manual or automated (Hartman and Howard, 1976). Personnel requests can be divided into 3 broad classes: Training, Regular, and Pre-emptive. Within each class numerical preference priority levels could be established by the scheduler. Only pre-emptive requests would be able to displace other requests within an existing schedule.

There are several ways in which priorities may be used within a scheduling system. In most cases priorities will be the cause of a schedule revision. On occasion, requests arrive

which are so important that they must be met immediately, for example emergency leave requests. In this type of situation scheduled personnel may need to be shuffled among duties in order to accommodate the temporary loss. Such situations have pre-emptive priority and they can severely disrupt normal operations (Hartman and Howard, 1976).

Another use of priorities is in the initial preparation of the master schedule. In cases of personnel requesting regular leave or attendance of educational classes, personnel may turn in duty preference dates for scheduled duty assignments. The scheduler will consider the requests and try to accommodate personnel preferences. Total requests may outweigh personnel resources in which the scheduler will determine what requests demand higher priority. A preferential priority ranking system is developed in order to simplify these subjective aspects.

#### **E. SCHEDULE EFFICIENCY**

When the total of all personnel requests is moderate almost any scheduling system will develop workable schedules. In periods of high demand, efficient scheduling may require a decrease in the amount of requests approved for personnel. Any proposed system should be tested under extreme conditions such as heavy personnel requests, many emergency leave requests, and operational requirements, for it is in these conditions that the differences between applicable systems will become apparent (Hartman and Howard, 1976).

#### **F. COMPARING ALTERNATIVE SCHEDULING SYSTEMS**

##### **1. Schedule Optimization**

A system which has more alternate choices available should be able to schedule personnel in a more efficient manner, thus leading to a schedule which is closer to optimal

(Poshyanonda, 1989). It follows that a scheduler which is expected to do more optimization will require more intricate decision logic to make the required choices between alternatives.

## **2. Priorities**

An important measure of effectiveness for scheduling system design is the extent to which the system can use priority and preference information in the scheduling process (Hartman and Howard, 1976). Different schedule alternatives are derived from the necessity to meet priorities of personnel.

## **3. Timing**

Timing issues are concerned with when and how often the schedule is produced and updated (Bow, 1966), and when personnel requests can be input into the system. The more convenient it is for a scheduler to produce output, the easier it will be to compare between alternatives. Additionally, it is easier to satisfy personnel requests if schedules are determined on the basis of planning by individuals submitting requests.

## **4. Stability and Changes**

All schedules are subject to changes for various reasons. If there is an instance of change, it is necessary to determine how the schedule shifts and its effect on personnel (Hartman and Howard, 1976). The scheduler needs to determine how to adjust in order to meet changing requirements.

## **5. Effect of the System for the User**

A variety of questions are considered. Is the system convenient to use and easy to understand? Can the system make use of expressed preferences? Will the system provide efficient schedules? Will the schedules be easier to produce and adjust?

## **G. CENTRALIZATION VS DECENTRALIZATION**

Manual systems tend to be centralized in the sense that there is an individual, the scheduler, who handles all requests, does the scheduling, and communicates with personnel. The scheduler is a focal point to whom the organization can go to for information about the system or to implement changes. Automated systems can be designed to be centralized with a designated scheduler maintaining the role of communication with both the computer and the personnel. Alternatively, in an automated system the role of scheduler may be abolished and personnel may communicate directly with the scheduling system using remote terminals.

## **H. DEGREE OF AUTOMATION FOR THE SCHEDULE**

Scheduling systems can range from fully manual to computer assisted to fully automated (Bow, 1966). The scheduling system consists of several functions, each of which can be performed in a variety of ways. These functions include the decision making function in which personnel are assigned to duty, the data storage and retrieval function which deals with such information as personnel data, requests, and priorities, and the decision function which produces the selected schedule (Hartman and Howard, 1976). Each of these functions can be done manually or by a computer. Hybrid systems in which some functions are automated and others are done manually should be considered as well as the extremes (Hartman and Howard, 1976).

### **III. REQUIREMENTS ANALYSIS**

This chapter discusses the watch scheduling process for the Navy, specifically focusing on FNMOC. Existing operational characteristics and problems of the scheduling procedures were investigated. Factors to be taken into consideration when designing the system include: human factors, social bias, and managerial concerns. In the data requirement and analysis phase, substantial information was gathered and analyzed.

The chapter is organized as follows. First, an overview of regulations regarding Navy watch organization is discussed, including experience based input for watch bill scheduling. The following section demonstrates what should be taken into consideration for developing any watch bill. The remainder of the chapter addresses FNMOC's current watch bill organization, and requirements it would incorporate into an automated system.

#### **A. WATCH ORGANIZATION**

In developing a watch rotation, the organization must first consider its operational requirements. The organization of watch sections are based upon the capabilities of personnel and the determination of the Commanding Officer. Based upon the requirements, the watch personnel functions and responsibilities are put into writing. The watch organization of the unit should be charted and explained to ensure all personnel are fully aware of their responsibilities and relationships while on watch (OPNAVINST 3120.32, 1994).

Currently, commands operate with schedules that are designed by hand. With manual methods, it is impossible to know the schedules are close to optimal in terms of serving personnel needs and meeting operational requirements. Evaluation of alternative schedules

and assignment of watch sections proves to be challenging with manual methods (Taylor & Huxley, 1989).

The generation of these watch bills manually is an extremely time consuming and frustrating process. Few individuals are skilled and effective at designing efficient watch bills. Experience from individuals involved with the generation of different types of watch bills demonstrates a great deal of factors are taken into consideration for designing a watch bill. These include:

- How many people are required to stand a specific watch?
- How many watches are needed to be filled?
- What is the duration of the watch?
- What are the personnel requirements to be qualified to stand a particular watch?
- How long does an individual train to be a qualified watch stander?
- What are the procedures for under-instruction watch standers?
- What is the expected time frame for personnel to qualify for watch positions?
- What personnel are available to stand a watch- who is on emergency/ regular leave, personnel who may be sick in quarters (SIQ), personnel training?
- How the watch sections are to be organized- strong individuals may be responsible for personnel under instruction for a position, personnel may need additional

attention to ensure duties are carried out.

- How will unforeseen circumstances affect the current watch bill, what modification may be in order?

Effective scheduling of the watch bill minimizes the shortages and surpluses of personnel, providing the highest correlation between number of watch standers needed and the number of watch standers actually on duty during the watch. The current use of manual scheduling causes the SWO to take all the possible watch considerations and balance among available personnel in order to produce a workable schedule. This system requires maintaining various files of personnel information including qualifications, expiration dates of qualifications such as CPR, leave requests, training dates, meetings personnel may be required to attend, collateral duties of personnel, and various information placing some emphasis on the watch bill generation. This manual system results in multiple stacks of paper files all subject to loss or damage. An advantage of an automated scheduling is that all the information having some impact on the schedule is maintained within the system and can be easily accessed. Additionally, it reduces variation in work load among personnel which helps boost morale (Taylor & Huxley, 1989).

Watch bills should not vary drastically between months. The existing watch bill of the ending month provides a starting point of the scheduling for the next month's watch bill for the SWO. Generating a schedule based on previous month eases the use of the system and matches the requirements for the new month in an efficient manner. Watch bill updating will



be a much easier task vice the manual system which does not provide simple updating procedures (Taylor& Huxley, 1989).

## **B. FNMOC REQUIREMENTS STUDY**

### **1. Watch Positions and Rotations**

All personnel assigned to a watch is responsible for ensuring the proper performance of all duties of the watch. Watch standers will receive sufficient training to perform the duties of the watch position prior to assignment. Personnel are fully attentive to the duties and responsibilities of the assigned watch position (OPNAVINST 3120.32c, 1994).

FNMOC has both military and civilian personnel as watch standers. Watch personnel are assigned in accordance with the organization shown in Figure 1.

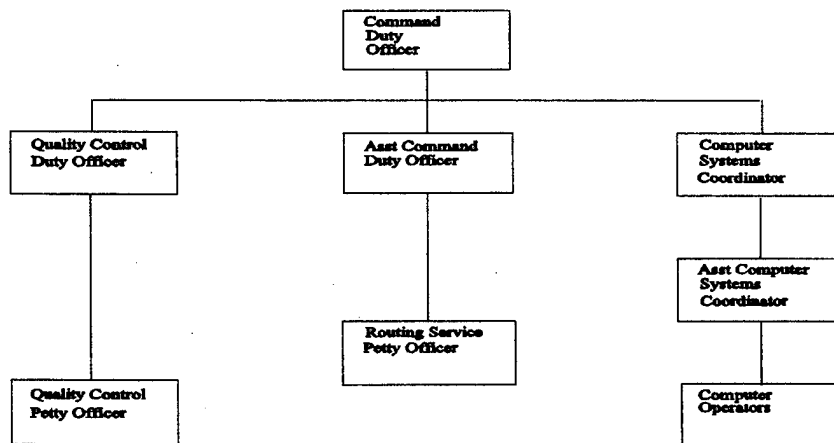


Figure 1

The military watch standers are partitioned into four sections. Each section is comprised of

the following positions: Command Duty Officer (CDO), Assistant Command Duty Officer (ACDO), Quality Control Duty Officer (QCDO), Quality Control Petty Officer (QCPO), and Routing Service Petty Officer (RSPO). The length of assignment to the watch is based on the conditions within the command (OPNAVINST 3120.32c, 1994). Each section is on watch for 12 hours. The day shift works from 0700-1800, and the Mid shift works from 1800- 0700.

The civilian watch standers are partitioned into three sections. Each section is comprised of the following positions: Computer Systems Coordinator (CSC), Assistant Computer Systems Coordinator (ACSC), and four Computer Operators (CO). Each section is on watch for eight hours: the Day shift works from 0700-1500, the Eve shift works from 1500-2100, and the Mid shift works from 2100-0700. Each section works a shift for four months and then rotates to the next section. The civilian scheduling aspect is not incorporated into the model at this time.

## **2. FNMOC Personnel**

FNMOC is composed of approximately 27 officers and 30 enlisted personnel available to stand watch. Personnel are divided into two categories: watch standers and day workers. Newly reporting military personnel are immediately assigned to the watch bill and work on the watch rotation for a minimum period of three months before shifting to their day working positions. Those individuals not currently on the watch rotation are maintained on the relief list which is used to fill vacancies on the watch bill due to sickness, emergencies, leave, Temporary Assigned Duty (TAD), and other unforeseen circumstances. The command employs approximately 50 civilians that work strictly on the watch rotation.

Personnel are assigned to the watch by the Command Watch Coordinator (CWC) who responsible to the Commanding Officer for the assignment and general supervision of permanent and relief watch standers. The CWC must prepare and promulgate the watch bill monthly, ensuring all positions are filled by qualified personnel and there is at least one individual with top secret clearance in each watch section.

#### **C. FNMOC CURRENT SCHEDULING SYSTEM**

Currently the task of watch bill preparation is performed manually by the CWC on a monthly basis. Department watch bill inputs are submitted to the CWC by the 24th of the preceding month. The CWC submits the completed watch bill via the Fleet Applications Department Head and the Director of Operations to the Executive Officer for approval (FNOC 3120.1, 1993).

The current manual scheduling procedures raises a number of areas of concern. First, the quality of watch bills produced depends on the skill of the individual scheduler, the CWC. Second, the informal scheduling methods are not easy to teach to a new scheduler in an expedient time frame. An automated system reduces the dependency on an individual's logic and allows for easy training to the follow-on CWC (FNOC 3120.1, 1993).

#### **D. FNMOC SYSTEM REQUIREMENTS**

The CWC requires a system that:

1. Generates optimal and realistic schedules that meet regulations using a microcomputer.
2. Permits easy adjustment of optimal watchbills to accommodate personnel without

sacrificing operations.

3. Has a user interface that is flexible and easy.
4. Maintains a list of daily required watches.
5. Maintains a database of personnel qualified to stand specific watches.
6. Assigns personnel to specific watches based on qualifications, clearance, and number of watches for the previous month.
7. Maintains a list of qualified relief watch standers.
8. Complies with Navy instructions governing watch standing.
9. Incorporates personnel preferences and exchanges of assigned watches.
10. Adapts to unique situations.

The scheduling system must accept the names of watch standing personnel, their qualifications, preference requests, and relief watch standers. The system must then assign personnel to appropriate duty rotations (FNOC 3120.1, 1993). It is most important that the watchbill is produced early enough to allow adequate time for preparation and planning by watch personnel.

Once the watchbill has been published, there must also be a facility to keep the schedule up to date. Changes in the watchbill may be required for a variety of reasons. Personnel assignments may change because of unforeseen circumstances such as emergency leave. Furthermore, a watch stander may be removed from the watch deck to fill a crucial day working position, or personnel may be removed from the watch deck as a result of punitive measures. Other changes may also result from incoming and departing organization personnel (FNOC 3120.1, 1993).



## IV. METHODOLOGY

This chapter addresses the design of the watch bill generator. The system requirements identified in the previous chapter are translated into a high level data model. Data modeling provides representation of objects about which data must be maintained, the characteristics of these objects, and the relationships between these objects. The two most used data modeling techniques are the semantic object model and the entity-relationship model. The conceptual data model for this thesis is the semantic object model.

The SALSA Recorder was used to develop the semantic object data models. This Windows based graphical data modeling tool is used to draw semantic object diagrams and generate the corresponding database schema.

The conceptual data model is next transformed into the logical design of the database and the practical design of the watch bill generator. The semantic object model of the watch bill generator is described in detail.

### A. SEMANTIC OBJECT MODELING

The semantic object model represents entities of interest as semantic objects within the user's domain and the relationships between these objects. A *semantic object* is a class of people, places, things, or transactions about which data is maintained. An object is a named collection of attributes that describes a distinct class of entities. These *attributes* describe characteristics or traits of the object being modeled.

#### 1. Attributes

There are four types of attributes that can be represented in semantic object modeling:

simple attributes, formula attributes, group attributes, and semantic object link attributes. A *simple attribute* is one that contains only one piece of data. For example, Social Security Number (SON) is a simple attribute of PERSONNEL as shown in figure 2. A *group attribute* is one that is composed of two or more attributes Personnel Name is an example of a group attribute and is composed of the two simple attributes FirstName and LastName. A *semantic object link attribute* is an attribute that consists of a semantic object. Using one object as an attribute of another object establishes a relationship between the objects. Snivel is a semantic object link between PERSONNEL object and SNIVEL object as shown in figures 2 and 3.

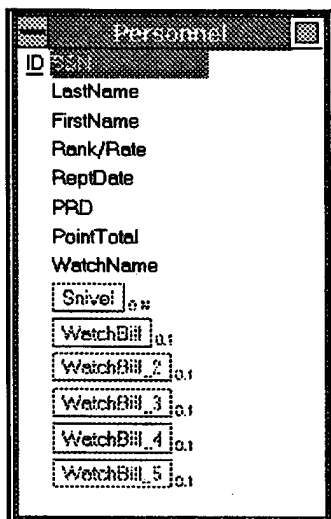


Figure 2

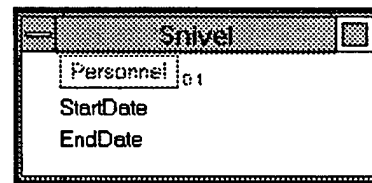


Figure 3

## 2. Instances

An *instance* of a semantic object is a member of the class described by the object. An instance of the PERSONNEL object might be 123-45-6789, Smith, John, LT, Oct 92, Oct 95, 34, CDO. An *identifier* is one or more attributes used to identify an instance of a semantic object. Identifiers may be unique or nonunique. In the PERSONNEL object, an

individual's SSN is assigned identifier because it will uniquely identify each person. SALSA depicts identifiers by the letters ID next to the object attribute.

A relationship is an association between objects. Relationships are described in terms of cardinality, which expresses how many instances of Attribute A can relate to one instance of Attribute B or Object B. Every attribute of an object has both minimum and maximum cardinality. The cardinality of an attribute reflects the rules governing the organization in which the data is used.

SALSA provides the capability to generate semantic object diagrams. These diagrams are used to summarize the structure of objects and present those structures visually as seen in APPENDIX A. The diagrams present a graphical representation of the objects, their attributes and identifiers, and their cardinalities (Dewitz and Olson, 1994).

## **B. CONCEPTUAL DESIGN**

### **1. Personnel Object**

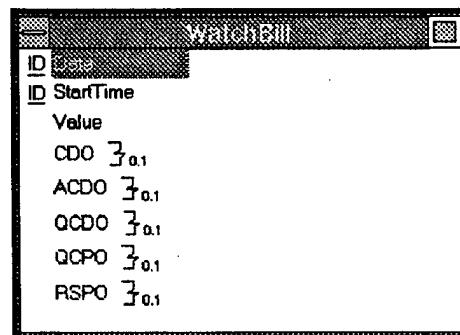
The PERSONNEL object describes each individual within the command. This is the central object for the entire data model. The attributes of PERSONNEL include each person's SSN, last name, first name, either rank or rate which is determined by whether the individual is an officer or enlisted, date individual reported to the command, projected rotation date (PRD- date upon which an individual is expected transfer from command), total of points earned for the month's watches, and the name of the watch position to which the individual is assigned (see figure 2). The object identifier is SSN because it is unique for each individual and allows user to easily distinguish between personnel. PERSONNEL is a *compound semantic object* which contains at least one semantic object link attribute.



PERSONNEL links to both the SNIVEL object and the WATCHBILL object. The information within both the SNIVEL and WATCHBILL objects relates to PERSONNEL. There is a many- to- many relationship between PERSONNEL and SNIVEL. Each person can have more than one snivel, and more than one snivel record can be assigned to one person. Similarly, PERSONNEL has a one-to -one relationship with WATCHBILL. One person can be assigned to each watch position, and each watch position can have only one person stand the watch.

## 2. WatchBill Object

The WATCHBILL is a *simple semantic object* that has no multi-valued attributes and no semantic object link attributes (see figure 4). The WATCHBILL object consists of the date of the watch, the start time of the watch, the point value for the watch, and the watch position. The identifier of the WATCHBILL is both the date and the start time because each day has a day watch and a mid watch, thus two different start times.



ID	StartTime	Value
CDO	3	0.1
ACDO	3	0.1
QCDO	3	0.1
QCPO	3	0.1
RSPO	3	0.1

Figure 4

### **3. Snivel Object**

The SNIVEL object records the requested time off by watch standers. Each SNIVEL covers one period of continuous requested time. A person may have more than one snivel for example, an individual may ask for every Tuesday off in January to attend school. This person would have four snivel records, one for each date. A person requesting 14 days leave would have one snivel record since it is one continuous period of time.

The SNIVEL object is a compound semantic object because it links with the PERSONNEL object (see figure 3). The SNIVEL object consists of the start and the end date of the period requested off. The object uses its link with PERSONNEL for identifier key of SSN.

### **4. Semantic Object Model Reports**

The semantic object reports are used to document each album which is a semantic object diagram and profile/attribute specification for a particular application. The reports are a valuable reference for the possibility of generating the scheduling database structure manually. The *Attributes report* (see APPENDIX A) lists all the album attributes in alphabetical order and indicates all attribute properties as well as the profile and profile-property defaults from which each attribute was derived. The *Profiles report* (see APPENDIX A) provides an alphabetical listing of all the profiles used in the album, including the profile properties.

## **C. DATA CONSIDERATIONS**

Preparation of the watchbill involves collecting information, rearranging the information in an acceptable form for the software and making a mathematical calculation of

watch values in order to determine how much each shift should be in order to operate efficiently. Preparing the watchbill takes expertise and judgement (Poshyanonda, 1989). The major elements that are involved in the watchbill scheduling system include a personnel data base, input data, and output reports.

### **1. Personnel Database**

A scheduling system needs to have access to certain data to perform the required scheduling task (Hartman and Howard, 1976). In a computer system for scheduling, various files must be created to maintain a list of all personnel and the watch positions to be filled. The data base maintains pertinent information concerning all organization personnel, i.e. name, rank, SSN, date reported, estimated departure date, watch qualifications completed, primary duty, and collateral duties. Each watch stander has a record containing required information with the individual's SSN as the primary key. The personnel database will be composed of all the records of individuals within the command.

### **2. Input Requirements**

The primary input for the scheduling system is the watch rotation to be filled and the personnel to fill the watches. Input includes personnel information from the personnel data base, required qualification, previous month's watch point total, and preference dates for leave, training, and school attendance. In order to determine supernumerary watch assignments, additional input information may be requested such as past watches stood, date rotated off the watch deck and date supernumerary personnel are unavailable.

### **3. Output Reports**

A wide variety of output reports may be produced by the scheduling system. Some

are provided in hard copy, while others are better displayed on computer screen. The primary output is a master schedule which gives an up-to-date list of scheduled times for all watch personnel and relief personnel.

Additionally, the scheduling system should be able to provide detailed information about any individual scheduled for watch, such as name, qualification, duty assigned, last watch stood, and number of watches for the month. A variety of summary statistics and management reports are prepared from the basic data within the system. Reports can be generated to produce a list of watch personnel and their qualifications, number of watches stood by each individual, and list of relief watch standers and their last watch duty.



## **V. IMPLEMENTATION**

This chapter discusses the transformation of the database design of the previous chapter into a working model of the watch bill generating system. An overview of the database software is presented first, followed by a description of the prototype including tables, queries, macros, modules, and forms.

### **A. DBMS SOFTWARE SELECTION**

A difficult aspect of developing the prototype was the final selection of what software to use to accomplish the task. The factors involved with the watchbill generating software included availability, cost, ease of use, on-line help, and simple programming. Additionally, the system had to satisfy the requirements FNMOC requested:

1. Generates optimal and realistic integer schedules with little difficulty.
2. Allows easy changes to schedules without needing to start over.
3. Provides flexible and easy to use interfaces.
4. Maintains up to the minute schedules.
5. Maintains an updated personnel database with specific information.
6. Compares information between different tables and records.
7. Adapts to unique situations.

The software choices available for implementation were Paradox, Dbase IV, Microsoft Access, and FoxPro. Access was chosen because it met the FNMOC requirements, and form and report generation is easy. Access also provided the ease of programming modules to

accomplish tasks for tailored information and the multi-user application feature allowing the database files to be shared on networks.

## **B. MICROSOFT ACCESS OVERVIEW**

Microsoft Access is a relational database management system for creating database applications for use with Windows 3.1. Access provides a database structure that combines related tables and their indexes, forms, reports, macros, and Basic code within a single .MDB database file (Jennings, 1994).

### **1. Database Tables**

A table is a collection of data about a subject. Data is displayed in tables in the row/column format similar to spreadsheet applications. The rows of a table are known as *records* and the columns are *fields*. A record contains information on a single object and a field contains the same type of information for all records in the table. The intersection of row and column is known as a *data cell*. Data within the data cells can be added or edited by positioning the mouse pointer at the point within the cell where the user wants the information and clicking the mouse.

The design mode for tables display the characteristics of each field in the table in a grid format similar to a spreadsheet. The Table Properties window allows the user to view and edit the properties that apply to the table object. This enables the user to enter a text description of the table and assign a value to the ValidationRule property of the table. A validation rule is an expression limiting the value that can be entered in a field establishing the data type of the information within the data cell. The Indexes window displays each field of the table and identifies the primary key field, the field or combination of fields that is used to

uniquely identify each record in the table. Indexes are created on specific fields to speed up sorting and searches on that field. Since Access is a relational database, the database management applications are capable of linking tables by key fields.

## **2. Queries**

Queries allow the database developer to determine how data is to be presented. The query facility allows questions about data to be formulated. The data needed to answer database questions may come from one or more tables, and the query culminates requested information together. The user must decide upon the criteria which must be met by the queried data items in order that they to be included in the display of the form or report.

## **3. Database Forms**

A form is a synonym for a user defined window in Access. These graphical forms create the user interface to the database tables, and they display data contained in tables and queries that enable the user to add new data and update or delete existing data. Additionally, they provide a convenient way for the user to view data in an organized and attractive manner. Multiple choice selections for fields can be created in order to use shorthand codes to represent a set of allowable values. A properly designed form facilitates data entry and minimizes input errors.

Forms are constructed from a collection of control objects that are individual design elements. These controls are the components that can be seen in the windows and dialogues boxes appearing in Access.

A new form can be created by clicking the Form button followed by New. A window is displayed for the user to select between using a Form Wizard or a Blank Form to design



the form. The Form Wizard prompts the user with questions about the form desired then builds the form based on the user's responses. The Form Wizard helps build single-columned forms, tabular forms, and forms with a sub-form or graph.

#### **4. Macros**

Access macros take the place of programming code required by other database applications, and make database applications more productive. Macros are lists of actions that are to occur in response to events. Some events result from user actions, such as clicking a command button, opening a table or form, or typing a value into a text box.

Actions for a macro are listed in the order of occurrence. The macro actions have easily identifiable names, such as OpenTable. Most actions require arguments that specify how the action works, such as RunCode which requires the programmer to provide the name of the function to be executed. The user sets the arguments to specify the performance of particular tasks.

Macro actions are entered in macro datasheets within the Macro Design window. A new macro database object is created each time the user writes and saves a macro. Clicking the Macro button followed by the New button creates the new macro. The series of actions to be carried out are indicated in the Actions box of the Macro window. A window associated with each selected action guides the user to input needed information for the macro to work properly. A paragraph to the right of the lower window informs the user the purpose of the action and what information is needed to fill the boxes.

#### **5. Modules**

Modules contain Access Basic code the user writes to perform operations that the

standard set of macros included in Access cannot support. Access Basic code may be used to create user-defined functions that substitute for complex expressions used repeatedly to validate data, compute values for text boxes, and perform other duties. The creation of a user defined function minimizes the potential typing errors and enables the user to document the way the expression works. Access Basic code may also be used to write expressions that include more complex decision structures than those provided, or to write expressions that need repetitive loops for operation.

Access Basic is a programming language. To execute the Access basic functions, the user should use Access' macro language or specify the name of a Basic function as the value of an event. Any macro action can be executed from Access basic by preceding the keyword with a **DoCmd** and a space.

## **6. Reports**

A report presents data on the printed page or the display screen. To create a report, the user clicks on the Report button followed by the New button. To assist the user in creating reports, Access provides a Report Wizard. The Report Wizard is similar to the Form Wizard in that it asks questions of the user about the report and creates the report based on the user's answers.

## **C. IMPLEMENTATION OF FNMOC WATCH BILL GENERATOR**

### **1. Table Creation**

The watch bill generator is based on three tables developed using Access. These tables are Personnel, Watch Bill, Snivel (see APPENDIX B). The entire system's database is built upon these three tables. Most data fields of the tables are text data types including

number fields such as SSN. This method bypasses the systems truncation of leading zeros and the maximum numerical value which may be exceeded by some SSNs.

***a. Personnel Table***

The Personnel Table is a listing of all personnel within FNMOC qualified for a specific watch position. The table is composed of 9 data fields including:

FIELD NAME	DATA TYPE	DOMAIN
● LastName	Text	Any last name
● FirstName	Text	Any valid first name
● SSN	Text	Series of 9 digits
● Rank/Rate	Text	Appropriate Navy rank/rate
● WatchName	Text	CDO, ACDO, QCDO, QCPO, RSPO
● ReptDate	Date/Time	Local date/time
● PRD	Date/Time	Local date/time
● PtsLastMonth	Number	Points accumulated last month
● PtsThisMonth	Number	Points accumulated this month

The data maintained in this table is the personal information of each individual within the command. WatchName may be one of 5 attribute domains: CDO, ACDO, QCDO, QCPO, or RSPO. ReptDate is the date the individual reported to the FNMOC, and PRD is the

individual's projected rotation date from FNMOC. PtsLastMonth accounts for the total points each individual accumulated by standing watches for the prior month. PtsThisMonth is a running tally of the points each individual is earning for watches of the current month.

Records are added to the Personnel Table by Add/Delete Personnel form. This form allows the user to add a new personnel record to the table, delete a personnel record, and return back to the Personnel Table upon completion.

***b. Watch Bill Table***

The Watch Bill Table is the table upon which the actual monthly watch bill is assigned. The table is composed of 8 data fields including:

FIELD NAME	DATA TYPE	DOMAIN
● StartDate	Date/Time	Local date/time
● StartTime	Date/Time	Local date/time
● WatchValue	Number	Points assigned to watch
● CDO_Personnel_ID_FK2	Text	SSN
● ACDO_Personnel_ID_FK3	Text	SSN
● QCDO_Personnel_ID_FK4	Text	SSN
● QCPO_Personnel_ID_FK5	Text	SSN
● RSPO_Personnel_ID_FK6	Text	SSN

The StartDate indicates the first day of the month for the watch bill. Each date will have 2 records assigned because there are two watches for each date. The StartTime is the time each

watch begins. The day watch starts at 0645 and the mid watch starts at 1845. Each watch is assigned a specific number of points in the WatchValue field. As personnel stand a watch, the value of that watch is added to their PtsThisMonth total. The last five fields indicate the watch position personnel are assigned. The SSN of individuals qualified for each position and assigned to that watch is recorded in the data cell. These fields are foreign keys which establish the relationships between the WatchBill table to the Personnel table.

*c. Snivel Table*

The Snivel Table is a listing of the dates and times individuals request not to be assigned watch. Snivels may result from leave, training, school, etc. Personnel are assigned snivel records for each continuous period of time off requested, an individual may have more than one snivel record. The Snivel Table consists of five data fields including:

FIELD NAME	DATA TYPE	DOMAIN
● Personnel_ID_FK1	Text	SSN
● LastName	Text	Any valid last name
● WatchName	Text	CDO, ACDO, QCDO, QCPO, RSPO
● StartDate	Date/Time	Local date/time
● EndDate	Date/Time	Local Date/time

The Personnel\_ID\_FK1 field is a foreign key to establish a relationship between the Snivel Table and the Personnel Table. The data is the SSN of personnel requesting time off. The StartDate indicates the day upon which an individual requests time off, and the EndDate is the date the individual will complete time off.

Records are added to the table through Snivel form. The form allows the user to Add/Delete a Snivel of an individual. The form records the person's last name, SSN, watch position, and start/end date of time off. Individuals may have more than one Snivel record within the Snivel Table.

## **2. System Queries**

Several specific queries were designed to sort data within the tables. The use of queries turns the data within the database into information useful to the user and are used to produce forms and reports of data stored within the database.

The PersonnelQuery produces a report of all the personnel within the command sorted by watch position and alphabetically within the list of personnel of each position. The Personnel Query can be broken down into various smaller query reports for each assignment, such as the CDOQuery, ACDOQuery, and QCPOQuery.

The user must set an assignment query of each watch position for each month. The query includes the StartDate, StartTime, WatchValue, PersonnelID, and LastName of individual. The assignment queries are to be programmed in prior to use of the system. The assignment queries are sorted in chronological order by the date.

In order to produce the watch bill, a query is used to develop the form for generating the watch bill. A monthly watch report will be generated for each watch position. The information is sorted by the point value of the watch, so the report does not assign watches in chronological order. Assignment is accomplished by the point value of the watch and the point total of the watch stander.

Each watch position has a snivel query associated with it, such as the

ACDOSnívelQuery. The snível queries breaks out personnel by their watch position and sorts them alphabetically. The break out of personnel sníveis by positions allows the user to easily view the assignment and necessary changing of personnel to insure the position is filled.

### **3. System Modules**

In order to tailor the database system to be a watch bill generator, some Access Basic code had to be developed. The code was written to accomplish tasks and then designed into macros to run the code. APPENDIX C contains the user designed functions and subfunctions for generating the watch bill information.

#### ***a. AssignWatchStanders***

The function AssignWatchStanders assigns watchstanders to the specified watch bill based on the watch stander's previous month's point total, value of the specific watch, and the individual's snível record. The purpose of this module is to take the list of available watches and list of available watch standers and assign personnel to watches. The function operates by opening the watch type form and the specified month sorted with the highest value watch listed first, and the list of personnel for that watch position sorted by the individual with the lowest point total for the previous month first. Before assigning an individual to a watch, the function goes to a subfunction CheckSnível to see if the person to be assigned has a snível contrasting the watch in question. Next the system checks the subfunction CheckIfNull to see if the watch has been filled or remains empty. Additionally, the system goes to the CheckSurroundDates subfunction to determine if the individual has been assigned the previous watch for a specific day. The system is not to assign personnel back to back watches, such as the day and mid watch on the same date, or the midwatch for

one day and the following date's day watch. If the watch is available, there is no snivel record for the individual, and the individual has not been assigned the watch prior, then the person is assigned to the watch. The system moves to the next watchstander to repeat the assignment loop.

***b. PreAssignWatchstanders***

This function allows the user to assign an individual to a specific watch prior to the system's schedule generating capability. The significance of this function is to allow personnel the ability to volunteer for certain watches such as holiday watches. The appropriate query is opened for the type of watch and the applicable month so the user is able to isolate the watch requested.

***c. AssignValueWithUserInput***

The purpose of this module is to allow the user to determine the value of the watches. The AssignValueWithUserInput function operates using the system's internal clock and calendar. The function opens the WatchBill table to determine the watch dates. The values for watches are broken down by day watches and mid watches for weekdays, and weekends. Weekend watches are worth more points than weekdays, and mid watches are worth more than day watches. This module allows the user to modify the values of the watches. There are default values built into the system if the user wishes not to modify. The default values are 1 point for day watch, weekday; 1.5 points for mid watch, weekday; 2 points for day watch , weekend; 2.5 points for mid watch, weekend.

***d. DeleteWatchstanders***

The purpose of this function is to delete watchstanders from the watch bill so



that a watchbill of a new month may be generated. The code allows the user to select the month to delete and each position that must be updated. The function requires the user to go into the watch bill form for the specific month and clears the information recorded within it. The point total for watchstander have been totaled prior to executing the function.

#### **4. System Forms**

The user interface for the watch bill generator is based upon various forms developed using Access. The forms are designed based upon queries from the database tables, and macros running Access code specifically written for the system. Forms are used to design a menu facility providing easy access for the user. Additionally, the user has the option to move back and forth between the forms allowing simple changing and updating procedures.

##### ***a. Personnel Forms***

The personnel forms are developed to provide the user the ability to add or delete records, update individual records, and view command personnel information at a simple glance from the Personnel Menu. The form design allows the user to click on View All Personnel button to see a listing of all personnel within the command. This provides a link to a follow- on form listing all command personnel and their information maintained within the database. This form provides the information to the user sorted by watch position. The user may select Add/Delete Record button which provides links to another form allowing the user the option to Add a New Record, Delete a Current Record, or Return to the Personnel Menu. Lastly, the Personnel Menu gives the user the option to view personnel sorted by the watch position for which they are qualified, such as the CDO form. The individual watch position forms provide an easy view of everyone qualified for that job and provides personnel

point totals. The watch position personnel forms allow the user to view a graph depicting how the personnel compare to one another in point totals for the previous month, and view reports of the same information.

***b. Snivel Forms***

The Snivel menu is developed into a form providing the user the ability to input personnel time-off requests according to the watch position they are assigned. The user selects the type of watch stander requesting the time off, and the system links to another form for that position. The Watch Stander Snivel form, such as the ACDO Snivels, allows the user to input snivels for personnel by their last name, and SSN. The user adds the period of time by StartDate and EndDate. The user is also provided the option to delete a snivel of an individual if a change arises. The Snivel form provides the user a view of all the snivels of personnel standing that watch position. This is useful for the user to eliminate possible conflicts in request dates.

***c. Generate Watchbill Forms***

The collection of Generate Watchbill forms provides the actual output of the watchbill for the month of choice. The forms are designed to allow the user to generate a separate watchbill for each watch position. The first form asks the user if a watchstander is to be pre-assigned to a specific watch. The user is provided the option to pre-assign or generate watchbill with no pre-assignments. The next form queries the user of what watchbill to generate, such as CDO watchbill. This form links to select month form allowing the user to determine what month the watchbill is applicable. Finally, the user is provided the Get Value form allowing modification to the values assigned to the watches. The form states the

default values assigned by the program, but value changes are allowable. With all the above information, the system will generate a watchbill for the watch standers to be output in a monthly watchbill form. The watchbill form lists for each watch position the watch date, time, value, watch stander's SSN, and last name.

## **5. View Watchbill**

The View Watchbill option allows the user to view the entire completed watchbill after it has been generated. Additionally, the forms provide the user a view of the watchbill after updates are incurred without having to go through the entire watchbill generating process again.

## **6. Reports**

Reports are used in the watchbill generator to provide the user with easy to access and read views of information related to the command and the watchbill system. A special form is designed to present the Reports Menu to the user. The menu allows the user to select the type of report they wish to generate. The choices are Personnel Reports, Snivel Reports, and Watchbill Reports. The reports option is provided in its own criteria of the system's main menu, but reports are available within the section they apply.

The Personnel Reports give the user the choices of selecting the type of report desired including: Command Personnel, CDOs, ACDOs, QCDOs, QCPOs, or RSPOs. The Personnel Report is a listing of all command personnel sorted by the watch position they are qualified to stand and listed alphabetically. The report provides a list of personnel by last name, first name, each individual's points for last month and the current total of points for the current month. The report calculates the total number of points earned by all personnel by

the watch qualification.

The reports generated for each watch position is similar to the Personnel report. The report provides a break out of data by the watch position so the user has an easier view of the information. The report lists the current date, position the report is applicable to, and the same information within the Personnel report, with the addition of the personnel SSNs.

The Snivel Report also allows the user to select type of report by the watch position of the individual requesting the snivel. The reports generated are categorized by the watch position, and provide a listing of the updated personnel requests for time off. The information in the report includes simply the last name of individual making the request, and the start and end date of the period of time.

The Watchbill Reports provides the user the ability to input the month and watch position required for the report. The system provides a report similar to the watchbill form. The report includes the current date, the watch position, and month of the watchbill. The output report provides the start date, start time, and last name of personnel assigned to the watch.



## **VI. CONCLUSION AND RECOMMENDATIONS**

In summary, this thesis has proven that automating a watchbill system is completely feasible and can be designed for all watchbills throughout the entire Navy. Each command could tailor its automated system to fulfill its own requirements, but the underlying system operations remain the same.

This chapter discusses the lessons learned and problems encountered in the process of developing the watchbill generator. Finally, recommendations of additional design capabilities are included for follow-on development.

### **A. LESSONS LEARNED**

The development of the prototype provided a venue of learning during the building process. The design of this current system requires the watchbill generator to be tailored to each specific command's individual watchbills. Further modifications to the system could implement greater flexibility allowing the system to be more generic to various types of watchbill scenarios.

The design process was made difficult by the necessity to develop 12 separate tables for each watch position for each month. Additionally, it was time consuming to have to design individual queries for each watch position. Along with needing development time for the separate tables and queries, the refinement of storage space increased dramatically. The system was designed as such due to developers limited knowledge of the software. Further use of Access may provide hints of better development strategies.

In some of the modules, such as the Pre-AssignWatchstanders, the user may be

required to input data while using the system. The user may be required to refer back to information stored within the database. It would have been helpful to design pull down menus with the needed information stored. For example, while pre-assigning a watchstander to a specific watch, a pull down menu listing all the watch positions, and the name and SSN of the personnel qualified to stand the watch would save the user time within the system.

Lastly, it would be easier for the user to have all the individual watchbills compiled into one full listing. It is possible to accomplish with the current prototype, but it requires a great deal of processing time.

## **B. PROBLEMS ENCOUNTERED**

Use of the DBMS Access provided a good working environment for developing the watchbill generator. It provided for ease of learning and use during prototype development. Additionally, Access provides the actual users an easy to use platform for developing the necessary watchbill. In the development phase, technical problems were encountered while using the software. The documentation accompanying Microsoft Access was unclear in some instances. Routine descriptions were ambiguous and minor setbacks occurred due to lack of knowledge. There should be more clarification within the software documentation. Though documentation should be written better, Microsoft maintains an Access help line available 24 hours a day. The help desk proved very useful for clarification. Lastly, Access does not have the capability of producing a calendar form. This would be useful for the user's view of the completed watchbills.

### **C. CONCLUSIONS**

This thesis proves it is feasible to design a completely automated schedule system using an existing DBMS package in order to replace a current manual process. The system developed for FNMOC is an over simplified example considering only five necessary watchbills. Various other fleet commands may have more dynamic environments requiring greater flexibility of the watchbill. A re-designed system would be needed to be able to meet all the needs of the SWOs within the fleet for every situation.

The prototype designed provides its users with a faster and more concise method for developing the watchbill. There are numerous benefits of the automated system. An increased saving of person-days for watchbill development every year since manual scheduling will no longer be necessary, and manual refinement will be eliminated. The model demonstrates the ability of generating alternate watchbills which increases the flexibility of the scheduling process. The system will always produce a watchbill that is complete, consistent, and satisfy the user requirements.

### **D. RECOMMENDATIONS**

Consideration for enhancing the developed prototype should be given. A few additional features not incorporated into the current system would provide the users an even better product. The development of a function to check for holidays is recommended. The system does not recognize dates as holidays. Holiday specifications will need to be programmed into the system with the use of Access basic code. Users can then provide personnel more points for standing a holiday watch.



A system redesign would allow the user to incorporate all the requirements that need to be considered for the scheduling of a watch to any other environment. At FNMOC the only requirements for watch scheduling are personnel qualifications, training requirements, and personnel availability. A SWO within the fleet has much more to consider including: Personnel Qualification Standards (PQS) completion, watch qualifications, training requirements, watch duration, lengthy under-instruction requirements, availability, the balance of watch personnel into various sections, the number of personnel qualified to fill one or more positions, inport requirements, underway requirements, and the confidence that the individual will complete the job. The automated system design can easily be altered to accommodate these additional requirements by developing specific modules to meet the need. The prototype system would benefit from a more modular design, and facilitate access to the database by other modules such as PQS or training management.

Lastly, the basic methodology of the prototype developed for FNMOC does not have to be restricted to primarily the watchbill. The design features can be manipulated to provide automation of additional Navy functions, such as PQS management, field day cleaning activities, equipment storage and check out procedures, and routine maintenance activities. An automated system provides satisfactory solutions in a short period of time. The usability of system can be expanded for planning and scheduling functions in any dynamic real-time environment.

**THIS PAGE IS INTENTIONALLY LEFT BLANK**



## **APPENDIX A. SEMANTIC OBJECT REPORTS**

# Semantic Object Report

Album: WATCH.ALB

---

## PERSONNEL Semantic Object

---

Caption:

Description:

---

### Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
SSN	Unique	1	1	Text	10	
LastName	None	1	1	Text	15	
FirstName	None	1	1	Text	15	
Rank/Rate	None	1	1	Text	10	
ReptDate	None	1	1	Date		
PRD	None	1	1	Date		
PointTotal	None	1	1	Text	10	
WatchName	None	0	1	Text	5	
Snivel	None	0	N (No Limit)	Semantic Object		
WatchBill	None	0	1	Semantic Object		
WatchBill_2	None	0	1	Semantic Object		
WatchBill_3	None	0	1	Semantic Object		
WatchBill_4	None	0	1	Semantic Object		
WatchBill_5	None	0	1	Semantic Object		

# Semantic Object Report

Album: WATCH.ALB

---

## SNIVEL Semantic Object

---

Caption:

Description:

---

### Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Personnel	None	0	1	Semantic Object		
StartDate	None	0	1	Date		
EndDate	None	0	1	Date		

# Semantic Object Report

Album: WATCH.ALB

## WATCHBILL Semantic Object

Caption:

Description:

### Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Date	Unique	1	1	Date		
StartTime	Unique	1	1	Text	10	
Value	None	1	1	Text	10	
CDO	None	0	1	Group		
Personnel	None	0	1	Semantic Object		
ACDO	None	0	1	Group		
Personnel	None	0	1	Semantic Object		
QCDO	None	0	1	Group		
Personnel	None	0	1	Semantic Object		
QCPO	None	0	1	Group		
Personnel	None	0	1	Semantic Object		
RSPO	None	0	1	Group		
Personnel	None	0	1	Semantic Object		

# Attribute Report

Album: WATCH.ALB

---

ACDO	Type: Group Profile: ACDO Contained in: WatchBill Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	Personnel
------	---	-----------------------	-----------

---

CDO	Type: Group Profile: CDO Contained in: WatchBill Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	Personnel
-----	--	-----------------------	-----------

---

Date	Type: Simple Value Profile: Date Contained in: WatchBill Caption: Description: ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Value Type: Date Length: Format: Initial Value:
------	--

---

EndDate	Type: Simple Value Profile: StartDate Contained in: Snivel Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: Initial Value:
---------	--

---

FirstName	Type: Simple Value Profile: FirstName Contained in: Personnel Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 15 Format: Initial Value:
-----------	--

---

LastName	Type: Simple Value Profile: LastName Contained in: Personnel Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 15 Format: Initial Value:
----------	---

---



# Attribute Report

Album: WATCH.ALB

---

Personnel	Type: Object Link Profile: Personnel Contained in: WatchBill.CDO Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-----------	--

---

Personnel	Type: Object Link Profile: Personnel Contained in: WatchBill.ACDO Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-----------	---

---

Personnel	Type: Object Link Profile: Personnel Contained in: WatchBill.QCDO Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-----------	---

---

Personnel	Type: Object Link Profile: Personnel Contained in: Snivel Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-----------	---

---

Personnel	Type: Object Link Profile: Personnel Contained in: WatchBill.QCPO Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-----------	---

---

Personnel	Type: Object Link Profile: Personnel Contained in: WatchBill.RSPO Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-----------	---

---

PointTotal	Type: Simple Value Profile: PointTotal Contained in: Personnel Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
------------	---

---

# Attribute Report

Album: WATCH.ALB

---

PRD  
Type: Simple Value  
Profile: PRD  
Contained in: Personnel  
Caption:  
Description:  
ID Status: None  
Minimum Required: 1  
Maximum Allowed: 1  
Value Type: Date  
Length:  
Format:  
Initial Value:

---

QCDO  
Type: Group  
Profile: QCDO  
Contained in: WatchBill  
Caption:  
Description:  
ID Status: None  
Minimum Required: 0  
Maximum Allowed: 1  
Minimum Count: 0  
Maximum Count: ALL  
Attributes Contained: Personnel

---

QCPO  
Type: Group  
Profile: QCPO  
Contained in: WatchBill  
Caption:  
Description:  
ID Status: None  
Minimum Required: 0  
Maximum Allowed: 1  
Minimum Count: 0  
Maximum Count: ALL  
Attributes Contained: Personnel

---

Rank/Rate  
Type: Simple Value  
Profile: Rank/Rate  
Contained in: Personnel  
Caption:  
Description:  
ID Status: None  
Minimum Required: 1  
Maximum Allowed: 1  
Value Type: Text  
Length: 10  
Format:  
Initial Value:

---

ReptDate  
Type: Simple Value  
Profile: ReptDate  
Contained in: Personnel  
Caption:  
Description:  
ID Status: None  
Minimum Required: 1  
Maximum Allowed: 1  
Value Type: Date  
Length:  
Format:  
Initial Value:

---

RSPO  
Type: Group  
Profile: RSPO  
Contained in: WatchBill  
Caption:  
Description:  
ID Status: None  
Minimum Required: 0  
Maximum Allowed: 1  
Minimum Count: 0  
Maximum Count: ALL  
Attributes Contained: Personnel

---

# Attribute Report

Album: WATCH.ALB

---

Snivel	Type: Object Link Profile: Snivel Contained in: Personnel Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)
--------	--

---

SSN	Type: Simple Value Profile: SSN Contained in: Personnel Caption: Description: ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
-----	--

---

StartDate	Type: Simple Value Profile: StartDate Contained in: Snivel Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: Initial Value:
-----------	--

---

StartTime	Type: Simple Value Profile: StartTime Contained in: WatchBill Caption: Description: ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
-----------	--

---

Value	Type: Simple Value Profile: Value Contained in: WatchBill Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
-------	--

---

WatchBill	Type: Object Link Profile: WatchBill Contained in: Personnel Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-----------	--

---

# Attribute Report

Album: WATCH.ALB

---

WatchBill_2	Type: Object Link Profile: WatchBill Contained in: Personnel Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-------------	--

---

WatchBill_3	Type: Object Link Profile: WatchBill Contained in: Personnel Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-------------	--

---

WatchBill_4	Type: Object Link Profile: WatchBill Contained in: Personnel Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-------------	--

---

WatchBill_5	Type: Object Link Profile: WatchBill Contained in: Personnel Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-------------	--

---

WatchName	Type: Simple Value Profile: WatchName Contained in: Personnel Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 5 Format: Initial Value:
-----------	---

---

# Profile Report

Album: WATCH.ALB

ACDO	Type: Group Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Format: Minimum Count: 0 Maximum Count: ALL	Profiles Contained:	Personnel
Address	Type: Group Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Format: Minimum Count: 0 Maximum Count: ALL	Profiles Contained:	Street City State Zip
AreaCode	Type: Simple Value Contained in: Phone Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 3 Format: Initial Value:		
CDO	Type: Group Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Format: Minimum Count: 0 Maximum Count: ALL	Profiles Contained:	Personnel
City	Type: Simple Value Contained in: Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 35 Format: Initial Value:		
CurrencyAmount	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Currency Length: Format: Initial Value:		

# Profile Report

## Album: WATCH.ALB

---

Date	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
------	--

---

Description	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Initial Value:
-------------	---

---

EventDate	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: Initial Value:
-----------	---

---

EventTime	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Time Length: Format: Initial Value:
-----------	---

---

FirstName	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 2 Maximum Allowed: 15 Value Type: Text Length: 15 Format: Initial Value:
-----------	---

---

Identifier-Numeric	Type: Simple Value Contained in: Caption: Description: ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Value Type: Long Integer Length: Format: Initial Value:
--------------------	---

---

# Profile Report

Album: WATCH.ALB

---

Identifier-Text	Type: Simple Value Contained in: Caption: Description: ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 15 Format: Initial Value:
<hr/>	
LastName	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 2 Maximum Allowed: 15 Value Type: Text Length: 15 Format: Initial Value:
<hr/>	
LocalNumber	Type: Simple Value Contained in: Phone Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 8 Format: Initial Value:
<hr/>	
MinRank/Rate	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
<hr/>	
Name	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
<hr/>	
Percent	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Floating Point Length: Format: Initial Value:

---

# Profile Report

## Album: WATCH.ALB

---

PersonName      Type: Simple Value  
                  Contained in:  
                  Caption:  
                  Description:  
                  ID Status: Non-unique  
                  Minimum Required: 1  
                  Maximum Allowed: 1  
                  Value Type: Text  
                  Length: 35  
                  Format:  
                  Initial Value:

---

Personnel        Type: Object Link  
                  Contained in: CDO, QCDO, ACDO, QCPO, RSPO  
                  Caption:  
                  Description:  
                  ID Status: None  
                  Minimum Required: 0  
                  Maximum Allowed: 1

---

---

Phone	Type: Group Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Format: Minimum Count: 0 Maximum Count: ALL	Profiles Contained:	AreaCode LocalNumber
-------	---	---------------------	-------------------------

---

Points            Type: Simple Value  
                  Contained in:  
                  Caption:  
                  Description:  
                  ID Status: None  
                  Minimum Required: 0  
                  Maximum Allowed: 1  
                  Value Type: Text  
                  Length: 10  
                  Format:  
                  Initial Value:

---

PointTotal        Type: Simple Value  
                  Contained in:  
                  Caption:  
                  Description:  
                  ID Status: None  
                  Minimum Required: 0  
                  Maximum Allowed: 1  
                  Value Type: Text  
                  Length: 10  
                  Format:  
                  Initial Value:

---

PRD                Type: Simple Value  
                  Contained in:  
                  Caption:  
                  Description:  
                  ID Status: None  
                  Minimum Required: 0  
                  Maximum Allowed: 1  
                  Value Type: Text  
                  Length: 10  
                  Format:  
                  Initial Value:

---



# Profile Report

Album: WATCH.ALB

---

QCDO	Type: Group Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Format: Minimum Count: 0 Maximum Count: ALL	Profiles Contained:	Personnel
------	---	---------------------	-----------

---

QCPO	Type: Group Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Format: Minimum Count: 0 Maximum Count: ALL	Profiles Contained:	Personnel
------	---	---------------------	-----------

---

Quantity	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Short Integer Length: Format: Initial Value:
----------	--

---

Rank/Rate	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 2 Maximum Allowed: 5 Value Type: Text Length: 10 Format: Initial Value:
-----------	--

---

ReptDate	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
----------	--

---

RSPO	Type: Group Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Format: Minimum Count: 0 Maximum Count: ALL	Profiles Contained:	Personnel
------	---	---------------------	-----------

---

Snivel	Type: Object Link Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
--------	--

---

# Profile Report

Album: WATCH.ALB

---

SSN	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
-----	--

---

StartDate	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
-----------	--

---

StartTime	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
-----------	--

---

State	Type: Simple Value Contained in: Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 2 Format: Initial Value:
-------	---

---

Street	Type: Simple Value Contained in: Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 30 Format: Initial Value:
--------	--

---

Value	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
-------	--

---

# Profile Report

Album: WATCH.ALB

---

WatchBill	Type: Object Link Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
-----------	--

---

WatchName	Type: Simple Value Contained in: Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
-----------	--

---

Zip	Type: Simple Value Contained in: Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 10 Format: Initial Value:
-----	--

---





## APPENDIX B. DATABASE TABLES

Table: Personnel									
ID	Last Name	First Name	SSN	Rank/Rate	Watch Name	Rept Date	PRD	Pst Last Month	Pst This Month
6	Whisenant	Larry	235-97-6543	03	QCPO	20-Mar-95	01-Oct-97	10.5	13
7	Buchanan	Todd	376-95-5606	03	ACDO	30-Jan-95	23-Aug-97	6	0
9	Benoit	Tom	467-21-2765	E3	RSPD	14-May-94	20-Jun-96	12	9.5
10	Smith	John	299-87-7766	03	CDO	02-Jul-93	30-Jun-96	8.5	8
11	Eleutero	Bill	342-95-2468	03	CDO	09-Oct-94	17-Dec-97	8	7.5
12	Moore	Dave	265-99-7433	01	CDO	03-Mar-94	21-Dec-96	10	7.5
13	Dashnerwertz	Joe	655-98-0077	01	ACDO	23-Feb-94	10-Mar-96	6.5	0
15	Mykyta	John	220-98-7764	03	CDO	12-Dec-93	15-Jun-95	8	8
16	Oschmann	Terry	367-94-6623	03	CDO	23-Sep-94	23-Sep-97	8.5	8
17	Warick	Freddy	233-98-7765	E1	RSPD	12-Feb-94	26-Mar-96	12	9.5
19	Bhargava	Hemant	330-98-7765	03	CDO	09-Jun-94	09-Jun-97	7.5	8.5
20	Jeffenes	Tom	112-87-5578	E7	QCDO	12-Feb-94	12-Feb-97	6.5	0
21	Shaffer	Scott	347-39-0087	E6	QCDO	30-Apr-94	30-Apr-97	6.5	0
22	Anderson	Roger	594-97-3761	E5	RSPD	10-Mar-95	10-Mar-98	10.5	10.5
23	Taylor	Todd	766-32-0942	E5	QCPO	30-Sep-94	30-Sep-97	10.5	13
24	Lefan	Rodney	478-22-6283	E5	RSPD	30-Jan-94	30-Jan-97	10.5	10.5
25	Kennemer	Glen	578-92-6073	E4	QCPO	01-Dec-93	01-Dec-96	11	12
27	Prentice	Larry	278-65-5590	E6	QCDO	30-Aug-94	30-Aug-97	6.5	0
28	Ward	Bill	982-45-0054	E6	QCDO	23-Sep-93	30-Sep-97	7.5	0
29	Grant	Tony	553-68-0129	E6	QCDO	30-Jan-94	31-Jan-97	8	0
30	Smith	Jason	498-65-6793	E5	QCDO	01-Dec-94	31-Dec-97	8	0
31	Wyatt	Derek	825-35-1724	E5	QCDO	01-Feb-94	30-Jan-97	8	0
32	Drake	Paul	176-58-9538	E6	QCDO	05-Nov-95	30-Nov-98	8	0
33	Crain	Leon	773-36-2850	E6	QCDO	14-Mar-95	31-Mar-98	8	0
34	Bird	Dave	276-65-0864	E6	QCDO	01-Jun-95	30-May-98	8	0
35	Senecal	Henrey	632-07-5621	E5	QCDO	25-Oct-95	31-Oct-98	8	0
36	Makowecki	Mike	452-87-6972	E6	QCDO	10-Apr-95	30-Apr-98	8	0
37	Loeffler	Susan	276-84-0087	01	ACDO	31-Jan-95	31-Jan-98	6.5	0
38	McDowell	Scott	496-22-2975	03	ACDO	01-Dec-94	30-Nov-97	7.5	0
39	Szczepanick	Stephanie	877-35-1952	01	ACDO	01-Feb-95	01-Feb-98	8	0
40	Kuypers	Mark	227-45-9786	03	ACDO	30-Jul-95	30-Jul-98	8	0
41	Garret	Dan	199-56-4489	03	ACDO	01-Mar-94	01-Mar-97	8	0
42	Whalen	Tim	387-32-7691	03	ACDO	01-Nov-94	30-Nov-97	8	0
43	Diminlou	Dominique	225-35-8745	03	ACDO	01-Apr-95	30-Apr-98	8	0
44	Ulises	Barry	664-97-6731	03	ACDO	24-Oct-95	31-Oct-98	8	0
45	Seibel	Scott	328-64-8834	01	ACDO	22-May-95	01-Jun-98	8	0
46	Mechels	Paul	764-29-7539	E6	ACDO	14-Sep-94	01-Sep-97	8.5	0
47	Smith	Peter	448-63-2297	03	CDO	26-Oct-95	30-Oct-98	8	8
48	Cannon	Philio	498-28-9610	03	CDO	12-Jan-96	30-Jan-99	8	8
49	Hudson	Thomas	660-35-6583	03	CDO	30-Nov-94	01-Dec-97	8	8
50	Corkhill	Hank	337-75-9942	04	CDO	23-Feb-94	01-Mar-97	8	8.5
51	Mays	Dave	638-76-5441	04	CDO	28-May-94	01-Jun-97	7.5	10
52	Spiker	Jon	482-94-3387	E5	QCPO	12-Jul-95	30-Jul-98	11.5	11

Personnel Table

Table: Snivel						
ID	Personnel ID	FK1	LastName	WatchName	StartDate	EndDate
1	342-96-2468		Eleuterio	CDO	07-Jan-96	09-Jan-96
2	285-99-7433		Moore	CDO	01-Feb-96	03-Feb-96
3	220-98-7764		Mykyta	CDO	24-Jan-96	28-Jan-96
4	482-94-3387		Spiker	QCPO	13-Feb-96	14-Feb-96
5	342-96-2468		Eleuterio	CDO	20-Feb-96	27-Feb-96
6	235-97-6543		Whisenhart	ACDO	11-Jan-96	17-Jan-96
8	376-95-5606		Buchanan	ACDO	10-Feb-96	15-Feb-96
9	399-11-9834		Stanberry	RSPD	20-Jan-96	26-Jan-96
10	467-21-2765		Benoit	RSPD	18-Feb-96	20-Feb-96
11	655-98-0077		Dashnerwert	ACDO	22-Jan-96	27-Jan-96
12	118-29-8390		Schneider	QCPO	02-Feb-96	05-Feb-96
13	773-36-2850		Crain	QCDO	08-Feb-96	09-Feb-96
14	982-45-0054		Ward	QCDO	22-Jan-96	30-Jan-96
15	227-45-9786		Kuypers	ACDO	02-Jan-96	08-Jan-96
* [Counter]						
Record: 1						

Snivel Table

Table: Watchbill									
ID	Start Date	End Date	Watch Name	EDU	Personnel	U	EDU	Personnel	U
1	01-Jan-96	06:45	1:496-28-9610		276-64-0087		112-87-5576		578-92-6073
2	01-Jan-96	18:45	1:5-636-76-5441		655-96-0077		982-45-0054		448-29-9631
3	02-Jan-96	06:45	1:387-94-6623		225-35-8745		773-36-2850		766-32-0942
4	02-Jan-96	18:45	1:5-342-96-2468		436-22-2975		553-68-0129		118-29-8390
13	03-Jan-96	06:45	1:239-87-7765		276-64-0087		982-45-0054		669-23-2267
14	03-Jan-96	18:45	1:5-220-98-7764		877-35-1952		498-65-6793		448-29-9631
15	04-Jan-96	06:45	1:342-96-2468		436-22-2975		553-68-0129		482-94-3387
16	04-Jan-96	18:45	1:5-337-75-9942		655-96-0077		347-39-0067		235-97-6543
17	05-Jan-96	06:45	1:285-98-7433		387-32-7631		176-58-9538		118-29-8390
18	05-Jan-96	18:45	2:5-367-94-6623		877-35-1952		498-65-6793		448-29-9631
19	06-Jan-96	06:45	2:2-220-98-7764		436-22-2975		553-68-0129		766-32-0942
20	06-Jan-96	18:45	2:5-330-98-7765		199-56-4489		825-35-1724		338-72-6795
21	07-Jan-96	06:45	2:367-94-6623		225-35-8745		773-36-2850		118-29-8390
22	07-Jan-96	18:45	2:5-448-63-2297		387-32-7631		176-58-9538		235-97-6543
23	08-Jan-96	06:45	1:220-98-7764		199-56-4489		825-35-1724		578-92-6073
24	08-Jan-96	18:45	1:5-330-98-7765		664-97-6731		276-65-0864		235-97-6543
25	08-Jan-96	06:45	1:5-636-76-5441		227-46-9786		498-65-6793		338-72-6795
26	08-Jan-96	18:45	1:5-498-28-9610		328-64-8834		632-07-5621		482-94-3387
27	10-Jan-96	06:45	1:342-96-2468		387-32-7631		176-58-9538		448-29-9631
28	10-Jan-96	18:45	1:5-448-63-2297		225-35-8745		773-36-2850		669-23-2267
69	11-Jan-96	06:45	1:330-98-7765						594-97-3761
70	11-Jan-96	18:45	1:5-660-35-6593		764-29-7539		492-87-6972		766-32-0942
71	12-Jan-96	06:45	1:5-636-76-5441		376-95-5606		982-45-0054		448-29-9631
72	12-Jan-96	18:45	2:5-337-75-9942		328-64-8834		632-07-5621		578-92-6073
73	13-Jan-96	06:45	2:330-98-7765		199-56-4489		825-35-1724		338-72-6795
74	13-Jan-96	18:45	2:5-660-35-6593		664-97-6731		276-65-0864		669-23-2267
75	14-Jan-96	06:45	2:448-63-2297		387-32-7631		176-58-9538		578-92-6073
76	14-Jan-96	18:45	2:5-498-28-9610		225-35-8745		773-36-2850		482-94-3387
77	15-Jan-96	06:45	1:220-98-7764		199-56-4489		825-35-1724		669-23-2267
78	15-Jan-96	18:45	1:5-330-98-7765		764-29-7539		492-87-6972		338-72-6795
79	16-Jan-96	06:45	1:448-63-2297		655-96-0077		347-39-0067		766-32-0942
80	16-Jan-96	18:45	1:5-498-28-9610		328-64-8834		632-07-5621		578-92-6073
81	17-Jan-96	06:45	1:660-35-6593		227-46-9786		553-68-0129		118-29-8390

Watchbill Table





## APPENDIX C. ACCESS BASIC CODE

Option Compare Database 'Use database order for string comparisons

Function AssignValueWithUserInput ()

\*\*\*\*\*

'This function assigns values to the watchbill based on the user's month  
'selection on the "Select Month" form, and values assigned on the "Get Value"  
'form.

\*\*\*\*\*

Dim db As Database, T As Table  
Set db = CurrentDB()

\*\*\*\*\*

'Get user's month selection.

\*\*\*\*\*

If Forms!SelectMonth![Month] = "1" Then  
Set T = db.OpenTable("Watchbill")  
ElseIf Forms!SelectMonth![Month] = "2" Then  
Set T = db.OpenTable("FebWatchbill")  
ElseIf Forms!SelectMonth![Month] = "3" Then  
Set T = db.OpenTable("MarWatchbill")  
ElseIf Forms!SelectMonth![Month] = "4" Then  
Set T = db.OpenTable("AprWatchbill")  
ElseIf Forms!SelectMonth![Month] = "5" Then  
Set T = db.OpenTable("MayWatchbill")  
ElseIf Forms!SelectMonth![Month] = "6" Then  
Set T = db.OpenTable("JunWatchbill")  
ElseIf Forms!SelectMonth![Month] = "7" Then  
Set T = db.OpenTable("JulWatchbill")  
ElseIf Forms!SelectMonth![Month] = "8" Then  
Set T = db.OpenTable("AugWatchbill")  
ElseIf Forms!SelectMonth![Month] = "9" Then  
Set T = db.OpenTable("SepWatchbill")  
ElseIf Forms!SelectMonth![Month] = "10" Then  
Set T = db.OpenTable("OctWatchbill")  
ElseIf Forms!SelectMonth![Month] = "11" Then  
Set T = db.OpenTable("NovWatchbill")

```

ElseIf Forms!SelectMonth![Month] = "12" Then
    Set T = db.OpenTable("DecWatchbill")
End If

```

```

*****
'Get values assigned by user from "Get Value" form.
*****

```

```

DoCmd OpenForm "GetValueForm"
T.MoveFirst
Do Until T.EOF
    T.Edit
    Let X = T!StartDate
    Let Y = T!StartTime
    DayNum = Weekday(X)

    If DayNum = 1 Or DayNum = 7 Then
        If Y = #06:45:00# Then
            T![WatchValue] = Forms!GetValueForm![Value2]
        Else
            T![WatchValue] = Forms!GetValueForm![Value3]
        End If
    ElseIf DayNum = 6 Then
        If Y = #06:45:00# Then
            T![WatchValue] = Forms!GetValueForm![Value]
        Else
            T![WatchValue] = Forms!GetValueForm![Value3]
        End If
    Else
        If Y = #06:45:00# Then
            T![WatchValue] = Forms!GetValueForm![Value]
        Else
            T![WatchValue] = Forms!GetValueForm![Value1]
        End If
    End If

    T.Update
    T.MoveNext
Loop
T.Close

End Function

```

Option Compare Database 'Use database order for string comparisons

### Function AssignWatchstanders ()

\*\*\*\*\*

'This function assigns watchstanders to the specified month's watchbill  
'based on the watchstander's previous month's point total, value of the  
'specific watch, and the individual's snivel record. After assigning an  
'individual to a watch, the function updates the individuals point total  
'for the current month and then moves to the next watchstander. The  
'process continues until all watches for the month have been assigned.

\*\*\*\*\*

Dim db As Database, Q As QueryDef, Q1 As QueryDef, dsWatches As Dynaset  
Dim dsPeople As Dynaset, dsSnivel As Dynaset, X, T As Table, SD, ED  
Dim Criteria, Criteria3, Criteria4, WBFULL, TOW, MTH, QN, FK  
Dim SW, Mark, Plus1, Minus1, IsAssigned

\*\*\*\*\*

'Get type of watchbill to be generated.

\*\*\*\*\*

Set db = CurrentDB()  
'DoCmd OpenForm "TypeOfWatchForm"  
'DoCmd OpenForm "SelectMonth"  
If Forms!TypeOfWatchForm![WatchType] = "1" Then  
    TOW = "CDO"  
    FK = "CDO\_Personnel\_ID\_FK2"  
ElseIf Forms!TypeOfWatchForm![WatchType] = "2" Then  
    TOW = "ACDO"  
    FK = "ACDO\_Personnel\_ID\_FK3"  
ElseIf Forms!TypeOfWatchForm![WatchType] = "3" Then  
    TOW = "QCDO"  
    FK = "QCDO\_Personnel\_ID\_FK4"  
ElseIf Forms!TypeOfWatchForm![WatchType] = "4" Then  
    TOW = "QCPO"  
    FK = "QCPO\_Personnel\_ID\_FK5"  
ElseIf Forms!TypeOfWatchForm![WatchType] = "5" Then  
    TOW = "RSPO"  
    FK = "RSPO\_Personnel\_ID\_FK6"  
End If

\*\*\*\*\*

'Get month and move points from "PointsThisMonth" to "PointsLastMonth" ..

\*\*\*\*\*

```
If Forms!SelectMonth![Month] = "1" Then
    MTH = "Jan"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "2" Then
    MTH = "Feb"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "3" Then
    MTH = "Mar"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "4" Then
    MTH = "Apr"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "5" Then
    MTH = "May"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "6" Then
    MTH = "Jun"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "7" Then
    MTH = "Jul"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "8" Then
    MTH = "Aug"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "9" Then
    MTH = "Sep"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "10" Then
    MTH = "Oct"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "11" Then
    MTH = "Nov"
    DoCmd RunMacro "DeletePtsAndMove"
ElseIf Forms!SelectMonth![Month] = "12" Then
    MTH = "Dec"
    DoCmd RunMacro "DeletePtsAndMove"
End If
```

\*\*\*\*\*

'Open the list of CDO Watches for the specified month, sorted with the highest

'value watch listed first, and the list of CDOs sorted with the lowest point  
'total from the previous month listed first.

\*\*\*\*\*

```
QN = MTH & TOW & "Watches"  
PQ = TOW & "Query"  
Set Q = db.OpenQueryDef(QN)  
Set dsWatches = db.CreateDynaset(QN)  
Set Q1 = db.OpenQueryDef(PQ)  
Set dsPeople = db.CreateDynaset(PQ)  
WBFull = 0  
dsPeople.MoveFirst
```

\*\*\*\*\*

'Begin main function.

\*\*\*\*\*

```
Do Until WBFull = 1  
    dsWatches.MoveFirst  
    Let X = dsPeople.SSN  
    Debug.Print X  
    IsAssigned = True  
    Do While IsAssigned = True  
        GoSub CheckIfNull  
        GoSub CheckSurroundDates  
        If IsAssigned = True Then  
            dsWatches.MoveNext  
        End If  
    Loop  
    GoSub CheckSnivel
```

\*\*\*\*\*

'If no surrounding watches or snivel records, assign person to that watch

\*\*\*\*\*

```
dsWatches.Edit  
dsWatches(FK) = X  
dsWatches.Update
```

\*\*\*\*\*

'Update Point Total for current month

\*\*\*\*\*

```

dsPeople.Edit
dsPeople.PtsThisMonth = dsPeople.PtsThisMonth + dsWatches.WatchValue
dsPeople.Update
dsPeople.MoveNext
If dsPeople.EOF Then
    DoCmd Requery Personnel
    dsPeople.MoveFirst
End If
Loop
dsWatches.Close
dsPeople.Close

```

\*\*\*\*\*

'CheckSurroundDates: This subroutine checks to make sure the person is not  
'already assigned to a watch on the current day, previous day, or following  
'day.

\*\*\*\*\*

CheckSurroundDates:

```

Let SW = dsWatches.StartDate
Mark = dsWatches.Bookmark      'Set bookmark
Plus1 = SW + 1
Criteria3 = "[StartDate] =#" & Plus1 & "#"
Minus1 = SW - 1
Criteria4 = "[StartDate] =#" & Minus1 & "#"
Criteria5 = "[StartDate] =#" & SW & "#"
Let IsAssigned = False
dsWatches.FindFirst Criteria3
Do Until (IsAssigned = True) Or (dsWatches.NoMatch)
    If dsWatches(FK) = X Then
        IsAssigned = True
        dsWatches.Bookmark = Mark      'Returns to original date
        Return
    End If
    dsWatches.FindNext Criteria3
Loop
dsWatches.FindFirst Criteria4
Do Until (IsAssigned = True) Or (dsWatches.NoMatch)
    If dsWatches(FK) = X Then
        IsAssigned = True
        dsWatches.Bookmark = Mark      'Returns to original date
        Return
    End If
    dsWatches.FindNext Criteria4
Loop

```

```

End If
dsWatches.FindNext Criteria4
Loop
dsWatches.FindFirst Criteria5
Do Until (IsAssigned = True) Or (dsWatches.NoMatch)
  If dsWatches(FK) = X Then
    IsAssigned = True
    dsWatches.Bookmark = Mark      'Returns to original date
    Return
  End If
  dsWatches.FindNext Criteria5
Loop
dsWatches.Bookmark = Mark          'Returns to original date

```

Return

```

*****
'CheckIfNull: This subroutine checks to see if the watch has been filled and
' continues until it reaches an empty watch.
*****

```

CheckIfNull:

```

If dsWatches.EOF Then GoSub ExitFunction
Do
  If Not IsNull(dsWatches(FK)) Then
    dsWatches.MoveNext
    If dsWatches.EOF Then
      WBFull = 1
      GoSub ExitFunction
    End If
  End If
Loop Until IsNull(dsWatches(FK))

```

Return

```

*****
'CheckSnivel Subroutine: Checks to see if person has a snivel during a
'particular date.
*****

```

CheckSnivel:



```

Set T = db.OpenTable("Snivel")
Set dsSnivel = db.CreateDynaset("Snivel")
FoundWatch = False
Criteria = "[Personnel_ID_FK1] = " & "" & X & ""
dsSnivel.FindFirst Criteria
If dsSnivel.NoMatch Then Return
Do Until FoundWatch = True
    Do Until dsSnivel.NoMatch
        Let SD = dsSnivel.StartDate
        Let ED = dsSnivel.EndDate
        If (dsWatches.StartDate >= SD) And (dsWatches.StartDate <= ED) Then
            dsWatches.MoveNext
            GoSub CheckIfNull
        Else
            FoundWatch = True
            dsSnivel.FindNext Criteria
        End If
    Loop
Loop
Return

*****
'Exit function
*****

ExitFunction:

    DoCmd OpenForm MTH & TOW & "Watchbill"
    DoCmd RepaintObject A_Form, MTH & TOW & "Watchbill"

End Function

```

Option Compare Database 'Use database order for string comparisons

Function DeletePtsAndMove ()

```
*****
'This function deletes previous month's total and replaces it with this
'month's total.
*****
    Dim db As Database, Q As QueryDef, dsPeople As Dynaset, X, TYP
    Set db = CurrentDB()

*****
'Get type of watchbill from the "Type of Watch" form.
*****

    If Forms!TypeOfWatchForm![WatchType] = "1" Then
        X = "CDO"
    ElseIf Forms!TypeOfWatchForm![WatchType] = "2" Then
        X = "ACDO"
    ElseIf Forms!TypeOfWatchForm![WatchType] = "3" Then
        X = "QCDO"
    ElseIf Forms!TypeOfWatchForm![WatchType] = "4" Then
        X = "QCPO"
    ElseIf Forms!TypeOfWatchForm![WatchType] = "5" Then
        X = "RSPO"
    ElseIf Forms!TypeOfWatchForm![WatchType] = "6" Then
        DoCmd Close A_Form, "TypeOfWatchForm"
    End If
    TYP = X & "Query"
    Set Q = db.OpenQueryDef(TYP)
    Set dsPeople = db.CreateDynaset(TYP)

*****
'Begin main function.
*****

    dsPeople.MoveFirst
    Do Until dsPeople.EOF
        dsPeople.Edit
        dsPeople.PtsLastMonth = dsPeople.PtsThisMonth
        dsPeople.PtsThisMonth = 0
        dsPeople.Update
        dsPeople.MoveNext
```

Loop  
dsPeople.Close

End Function

Option Compare Database 'Use database order for string comparisons

Function DeleteWatchstanders ()

\*\*\*\*\*

'This function deletes watchstanders from the watchbill so a new watchbill  
'may be generated.

\*\*\*\*\*

Dim db As Database, T As Table, X

Set db = CurrentDB()

\*\*\*\*\*

'Get month of new watchbill from "Select Month" form.

\*\*\*\*\*

```
If Forms!SelectMonth![Month] = "1" Then
    Set T = db.OpenTable("Watchbill")
ElseIf Forms!SelectMonth![Month] = "2" Then
    Set T = db.OpenTable("FebWatchbill")
ElseIf Forms!SelectMonth![Month] = "3" Then
    Set T = db.OpenTable("MarWatchbill")
ElseIf Forms!SelectMonth![Month] = "4" Then
    Set T = db.OpenTable("AprWatchbill")
ElseIf Forms!SelectMonth![Month] = "5" Then
    Set T = db.OpenTable("MayWatchbill")
ElseIf Forms!SelectMonth![Month] = "6" Then
    Set T = db.OpenTable("JunWatchbill")
ElseIf Forms!SelectMonth![Month] = "7" Then
    Set T = db.OpenTable("JulWatchbill")
ElseIf Forms!SelectMonth![Month] = "8" Then
    Set T = db.OpenTable("AugWatchbill")
ElseIf Forms!SelectMonth![Month] = "9" Then
    Set T = db.OpenTable("SepWatchbill")
ElseIf Forms!SelectMonth![Month] = "10" Then
    Set T = db.OpenTable("OctWatchbill")
ElseIf Forms!SelectMonth![Month] = "11" Then
    Set T = db.OpenTable("NovWatchbill")
ElseIf Forms!SelectMonth![Month] = "12" Then
    Set T = db.OpenTable("DecWatchbill")
End If
```

T.MoveFirst

```
Do Until T.EOF
    T.Edit
```

```
*****
```

```
'Get type of watchbill to be generated from "Type Of Watch" form.
```

```
*****
```

```
    If Forms!TypeOfWatchForm![WatchType] = "1" Then
        Let T!CDO_Personnel_ID_FK2 = X
    ElseIf Forms!TypeOfWatchForm![WatchType] = "2" Then
        Let T!ACDO_Personnel_ID_FK3 = X
    ElseIf Forms!TypeOfWatchForm![WatchType] = "3" Then
        Let T!QCDO_Personnel_ID_FK4 = X
    ElseIf Forms!TypeOfWatchForm![WatchType] = "4" Then
        Let T!QCPO_Personnel_ID_FK5 = X
    ElseIf Forms!TypeOfWatchForm![WatchType] = "5" Then
        Let T!RSPO_Personnel_ID_FK6 = X
    ElseIf Forms!TypeOfWatchForm![WatchType] = "6" Then
        DoCmd Close A_Form, "TypeOfWatchForm"
    End If
```

```
*****
```

```
'Begin main function
```

```
*****
```

```
    If Not IsNull(X) Then
        X = ""
    End If
    T.Update
    T.MoveNext
Loop
T.Close
```

```
End Function
```

Option Compare Database 'Use database order for string comparisons

Function ShowMonthForm ()

\*\*\*\*\*

'This function opens the particular "Snivel" form based on user input on the  
'"Select Month" form.

\*\*\*\*\*

Dim db As Database  
Set db = CurrentDB()

\*\*\*\*\*

'Get type of watch from the Select Month form

\*\*\*\*\*

DoCmd OpenForm "SelectMonth"  
If Forms!SelectMonth![Month] = "CDO" Then  
    DoCmd OpenForm "CDOSnivels"  
ElseIf Forms!SnivelMenu![WatchType] = "ACDO" Then  
    DoCmd OpenForm "ACDOSnivels"  
ElseIf Forms!SnivelMenu![WatchType] = "QCDO" Then  
    DoCmd OpenForm "QCDOSnivels"  
ElseIf Forms!SnivelMenu![WatchType] = "QCPO" Then  
    DoCmd OpenForm "QCPOSnivels"  
ElseIf Forms!SnivelMenu![WatchType] = "RSPO" Then  
    DoCmd OpenForm "RSPOSnivels"  
End If

End Function

Option Compare Database 'Use database order for string comparisons

Function ShowPersonnelReports ()

\*\*\*\*\*

'This function opens the appropriate "Snivel" report based on user input on  
'the "Personnel Reports" form.

\*\*\*\*\*

Dim db As Database

Set db = CurrentDB()

DoCmd OpenForm "PersonnelReports"

If Forms!PersonnelReports![ReportType] = "1" Then

DoCmd OpenReport "PersonnelReport", A\_PREVIEW

ElseIf Forms!PersonnelReports![ReportType] = "2" Then

DoCmd OpenReport "CDOSnivels", A\_PREVIEW

ElseIf Forms!PersonnelReports![ReportType] = "3" Then

DoCmd OpenReport "ACDOSnivels", A\_PREVIEW

ElseIf Forms!PersonnelReports![ReportType] = "4" Then

DoCmd OpenReport "QCDOSnivels", A\_PREVIEW

ElseIf Forms!PersonnelReports![ReportType] = "5" Then

DoCmd OpenReport "QCPOSnivels", A\_PREVIEW

ElseIf Forms!PersonnelReports![ReportType] = "6" Then

DoCmd OpenReport "RSPOSnivels", A\_PREVIEW

End If

End Function

Option Compare Database 'Use database order for string comparisons

Function ShowPersonnelForms ()

\*\*\*\*\*

'This function opens the appropriate personnel form based on user input on  
'the "Personnel Menu" form.

\*\*\*\*\*

```
Dim db As Database
Set db = CurrentDB()
DoCmd OpenForm "PersonnelMenu"
If Forms!PersonnelMenu![WatchType] = "1" Then
    DoCmd OpenForm "CDOForm"
ElseIf Forms!PersonnelMenu![WatchType] = "2" Then
    DoCmd OpenForm "ACDOForm"
ElseIf Forms!PersonnelMenu![WatchType] = "3" Then
    DoCmd OpenForm "QCDOForm"
ElseIf Forms!PersonnelMenu![WatchType] = "4" Then
    DoCmd OpenForm "QCPOForm"
ElseIf Forms!PersonnelMenu![WatchType] = "5" Then
    DoCmd OpenForm "RSPOForm"
End If
```

End Function



Option Compare Database 'Use database order for string comparisons

Function ShowSnivelReports ()

\*\*\*\*\*

'This function opens the appropriate "Snivel" report based on user input on  
'the "Snivel Reports" form.

\*\*\*\*\*

```
Dim db As Database
Set db = CurrentDB()
DoCmd OpenForm "SnivelReports"
If Forms!SnivelReports![ReportType] = "1" Then
    DoCmd Close A_FORM, "SnivelReports"
ElseIf Forms!SnivelReports![ReportType] = "2" Then
    DoCmd OpenReport "CDOSnivels", A_PREVIEW
ElseIf Forms!SnivelReports![ReportType] = "3" Then
    DoCmd OpenReport "ACDOSnivels", A_PREVIEW
ElseIf Forms!SnivelReports![ReportType] = "4" Then
    DoCmd OpenReport "QCDOSnivels", A_PREVIEW
ElseIf Forms!SnivelReports![ReportType] = "5" Then
    DoCmd OpenReport "QCPOSnivels", A_PREVIEW
ElseIf Forms!SnivelReports![ReportType] = "6" Then
    DoCmd OpenReport "RSPOSnivels", A_PREVIEW
End If
```

End Function

Option Compare Database 'Use database order for string comparisons

Function ShowSnivelForms ()

\*\*\*\*\*

'This function opens the appropriate "Snivel" form based on user input on the  
'"Snivel Menu" form.

\*\*\*\*\*

```
Dim db As Database
Set db = CurrentDB()
DoCmd OpenForm "SnivelMenu "
If Forms!SnivelMenu![WatchType] = "1" Then
    DoCmd OpenForm "CDOSnivels"
ElseIf Forms!SnivelMenu![WatchType] = "2" Then
    DoCmd OpenForm "ACDOSnivels"
ElseIf Forms!SnivelMenu![WatchType] = "3" Then
    DoCmd OpenForm "QCDOSnivels"
ElseIf Forms!SnivelMenu![WatchType] = "4" Then
    DoCmd OpenForm "QCPOSnivels"
ElseIf Forms!SnivelMenu![WatchType] = "5" Then
    DoCmd OpenForm "RSPOSnivels"
ElseIf Forms!SnivelMenu![WatchType] = "6" Then
    DoCmd Close A_Form, "SnivelMenu"
```

End If

End Function

Option Compare Database 'Use database order for string comparisons

Function ShowWatchbillReports ()

\*\*\*\*\*

'This function opens the appropriate "Watchbill" report based on user input  
'on the "Watchbill Reports" form.

\*\*\*\*\*

Dim db As Database, WT  
Set db = CurrentDB()

\*\*\*\*\*

'Get type of watch

\*\*\*\*\*

DoCmd OpenForm "WatchbillReports"  
If Forms!WatchbillReports![WatchName] = "CDO" Then  
    WT = "CDO"  
ElseIf Forms!WatchbillReports![WatchName] = "ACDO" Then  
    WT = "ACDO"  
ElseIf Forms!WatchbillReports![WatchName] = "QCDO" Then  
    WT = "QCDO"  
ElseIf Forms!WatchbillReports![WatchName] = "QCPO" Then  
    WT = "QCPO"  
ElseIf Forms!WatchbillReports![WatchName] = "RSPO" Then  
    WT = "RSPO"  
End If

\*\*\*\*\*

\*\*\*\*

'Get month

\*\*\*\*\*

\*\*\*\*

If Forms!WatchbillReports![Month] = "1" Then  
    MTH = "Jan"  
ElseIf Forms!WatchbillReports![Month] = "2" Then  
    MTH = "Feb"  
ElseIf Forms!WatchbillReports![Month] = "3" Then  
    MTH = "Mar"  
ElseIf Forms!WatchbillReports![Month] = "4" Then  
    MTH = "Apr"

```
ElseIf Forms!WatchbillReports![Month] = "5" Then
    MTH = "May"
ElseIf Forms!WatchbillReports![Month] = "6" Then
    MTH = "Jun"
ElseIf Forms!WatchbillReports![Month] = "7" Then
    MTH = "Jul"
ElseIf Forms!WatchbillReports![Month] = "8" Then
    MTH = "Aug"
ElseIf Forms!WatchbillReports![Month] = "9" Then
    MTH = "Sep"
ElseIf Forms!WatchbillReports![Month] = "10" Then
    MTH = "Oct"
ElseIf Forms!WatchbillReports![Month] = "11" Then
    MTH = "Nov"
ElseIf Forms!WatchbillReports![Month] = "12" Then
    MTH = "Dec"
End If
```

```
RN = MTH & WT & "Watchbill"
DoCmd OpenReport (RN), A_PREVIEW
```

```
End Function
```

Option Compare Database 'Use database order for string comparisons

Function ShowWatchbillForms ()

\*\*\*\*\*

'This function displays the appropriate watchbill form

\*\*\*\*\*

Dim db As Database, WT

Set db = CurrentDB()

\*\*\*\*\*

'Get type of watchbill from the View Watchbills form

\*\*\*\*\*

DoCmd OpenForm "ViewWatchbills"

If Forms!ViewWatchbills![WatchName] = "CDO" Then

WT = "CDO"

ElseIf Forms!ViewWatchbills![WatchName] = "ACDO" Then

WT = "ACDO"

ElseIf Forms!ViewWatchbills![WatchName] = "QCDO" Then

WT = "QCDO"

ElseIf Forms!ViewWatchbills![WatchName] = "QCPO" Then

WT = "QCPO"

ElseIf Forms!ViewWatchbills![WatchName] = "RSPO" Then

WT = "RSPO"

End If

\*\*\*\*\*

'Get month from the View Watchbills form

\*\*\*\*\*

If Forms!ViewWatchbills![Month] = "1" Then

MTH = "Jan"

ElseIf Forms!ViewWatchbills![Month] = "2" Then

MTH = "Feb"

ElseIf Forms!ViewWatchbills![Month] = "3" Then

MTH = "Mar"

ElseIf Forms!ViewWatchbills![Month] = "4" Then

MTH = "Apr"

ElseIf Forms!ViewWatchbills![Month] = "5" Then

MTH = "May"

ElseIf Forms!ViewWatchbills![Month] = "6" Then

```
    MTH = "Jun"
ElseIf Forms!ViewWatchbills![Month] = "7" Then
    MTH = "Jul"
ElseIf Forms!ViewWatchbills![Month] = "8" Then
    MTH = "Aug"
ElseIf Forms!ViewWatchbills![Month] = "9" Then
    MTH = "Sep"
ElseIf Forms!ViewWatchbills![Month] = "10" Then
    MTH = "Oct"
ElseIf Forms!ViewWatchbills![Month] = "11" Then
    MTH = "Nov"
ElseIf Forms!ViewWatchbills![Month] = "12" Then
    MTH = "Dec"
End If

RN = MTH & WT & "Watchbill"
DoCmd OpenForm (RN)

End Function
```

Option Compare Database 'Use database order for string comparisons

Function TypeOfWatchOptions ()

\*\*\*\*\*

'This function deletes watchstanders from the previous watchbill and calls  
'the macro that assigns new watchstanders.

\*\*\*\*\*

```
Dim db As Database
Set db = CurrentDB()
DoCmd OpenForm "TypeOfWatchForm"
If Forms!TypeOfWatchForm![WatchType] = "6" Then
    DoCmd Close A_Form, "TypeOfWatchForm"
Else
    DoCmd RunMacro "Delete/AssignWatchstanders"
End If
```

End Function

Option Compare Database 'Use database order for string comparisons

Function PreAssignWatchstanders ()

\*\*\*\*\*

'This function opens the appropriate query so that a watchstander can be  
'pre-assigned.

\*\*\*\*\*

Dim db As Database, Q As QueryDef, dsWatches As Dynaset, TOW, FK  
Set db = CurrentDB()

\*\*\*\*\*

'Get type of watch from the Type Of Watch Form

\*\*\*\*\*

If Forms!TypeOfWatchForm![WatchType] = "1" Then  
    TOW = "CDO"  
    FK = "CDO\_Personnel\_ID\_FK2"  
ElseIf Forms!TypeOfWatchForm![WatchType] = "2" Then  
    TOW = "ACDO"  
    FK = "ACDO\_Personnel\_ID\_FK3"  
ElseIf Forms!TypeOfWatchForm![WatchType] = "3" Then  
    TOW = "QCDO"  
    FK = "QCDO\_Personnel\_ID\_FK4"  
ElseIf Forms!TypeOfWatchForm![WatchType] = "4" Then  
    TOW = "QCPO"  
    FK = "QCPO\_Personnel\_ID\_FK5"  
ElseIf Forms!TypeOfWatchForm![WatchType] = "5" Then  
    TOW = "RSPO"  
    FK = "RSPO\_Personnel\_ID\_FK6"  
ElseIf Forms!TypeOfWatchForm![WatchType] = "6" Then  
    DoCmd Close A\_Form, "TypeOfWatchForm"  
End If

\*\*\*\*\*

'Get month

\*\*\*\*\*

If Forms!SelectMonth![Month] = "1" Then  
    MTH = "Jan"  
ElseIf Forms!SelectMonth![Month] = "2" Then  
    MTH = "Feb"



```

ElseIf Forms!SelectMonth![Month] = "3" Then
    MTH = "Mar"
ElseIf Forms!SelectMonth![Month] = "4" Then
    MTH = "Apr"
ElseIf Forms!SelectMonth![Month] = "5" Then
    MTH = "May"
ElseIf Forms!SelectMonth![Month] = "6" Then
    MTH = "Jun"
ElseIf Forms!SelectMonth![Month] = "7" Then
    MTH = "Jul"
ElseIf Forms!SelectMonth![Month] = "8" Then
    MTH = "Aug"
ElseIf Forms!SelectMonth![Month] = "9" Then
    MTH = "Sep"
ElseIf Forms!SelectMonth![Month] = "10" Then
    MTH = "Oct"
ElseIf Forms!SelectMonth![Month] = "11" Then
    MTH = "Nov"
ElseIf Forms!SelectMonth![Month] = "12" Then
    MTH = "Dec"
End If

```

```

QN = MTH & TOW & "Watches"
Set Q = db.OpenQueryDef(QN)
Set dsWatches = db.CreateDynaset(QN)
DoCmd OpenQuery QN

```

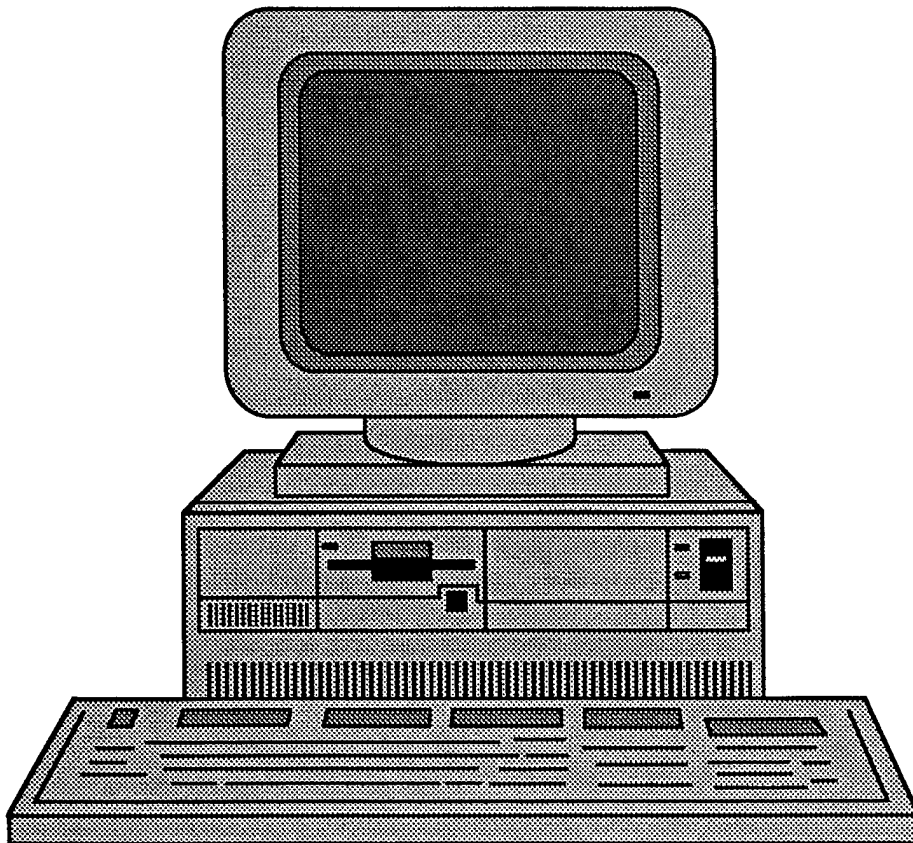
End Function





**APPENDIX D. USER'S MANUAL**

**FNMOCAUTOMATED  
WATCHBILL GENERATOR'S  
REFERENCE GUIDE**



**USER'S MANUAL**

## **A. INSTALLATION**

### **1. System Requirements**

The FNMOC Automated Watchbill Generator (FNMOCAUTWATGEN) is designed using Microsoft Access. To use FNMOCAUTWATGEN and Microsoft Access you need the following:

- IBM-compatible personal computer with an 80386 or higher processor.
- Hard disk with 13 megabytes of free space.
- Mouse or other compatible pointing device.
- EGA, VGA, or compatible display.
- Two megabytes of random-access memory.
- MS-DOS version 3.1 or later, and Microsoft Windows version 3.0 or later.

### **2. Installation Disks**

In addition to the disks for Microsoft Access, one disk is required. FNMOCAUTWATGEN is contained in the file db2.mdb. To open the FNMOCAUTWATGEN database:

- Start Microsoft Access
- From the File menu, choose Open Database.
- In the File Name box, select db2.mdb.
- Choose the OK button.

Microsoft Access opens the FNMOCAUTWATGEN database and displays the Main Menu form.

### **3. Backing Up the Database**

Creating backup copies is one of the most important database administration tasks. Backups protect you against loss of data and the information that describes your data. If you make unwanted changes to the database which can't be reversed, you can always return to the most recent backup copy. To back up the database:

1. Close the database.
2. Using the Windows File Manager, backup software, or the MS-DOS copy command,

copy the Microsoft Access .MDB file to the backup system of your choice.

To restore the database from the backup copy: use the Windows File Manager, backup software, or the MS-DOS copy command, copy the backup .MDB file to your database directory.

Make backup copies on a regular basis and keep backup copies of your data off site in case of fire, flood or other natural disaster.

## **B. OVERVIEW**

The FNMOCAUTWATGEN is designed to implement equality in watch standing. Using the program, each watch is worth a specific number of points; the watches that everyone wants (like those during the work week) are worth a lower value than the watches you have to pay someone to stand (like Christmas watches). For example, a week day watch may be worth 1 point while a weekend night watch may be worth 2.5 points and a holiday day watch may be worth 4 points. Recommended values are programmed into the system, but the system allows you to tailor the values for your specific needs.

All watchstanders have two point totals: "PtsThisMth" and "PtsLastMth." A watchstander's "PtsThisMth" total is increased each month by the value of the watches they were assigned that month. Before next month's watchbill is generated, the watchstander's "PtsThisMth" total is shifted to his or her's "PtsLastMth" total, and the next month's watchbill is created based on the previous month's assignments.

Prior to assigning personnel to watches, the program looks at the value of all the watches for the month selected, and sorts them so that the highest value watches are assigned first. The program then looks at all personnel qualified to stand the watch, and sorts them so that the person with the lowest "PtsLastMth" total is assigned the highest

value watch. In theory, the person with the lowest point total from the previous month, will be assigned the highest value (worst) watch.

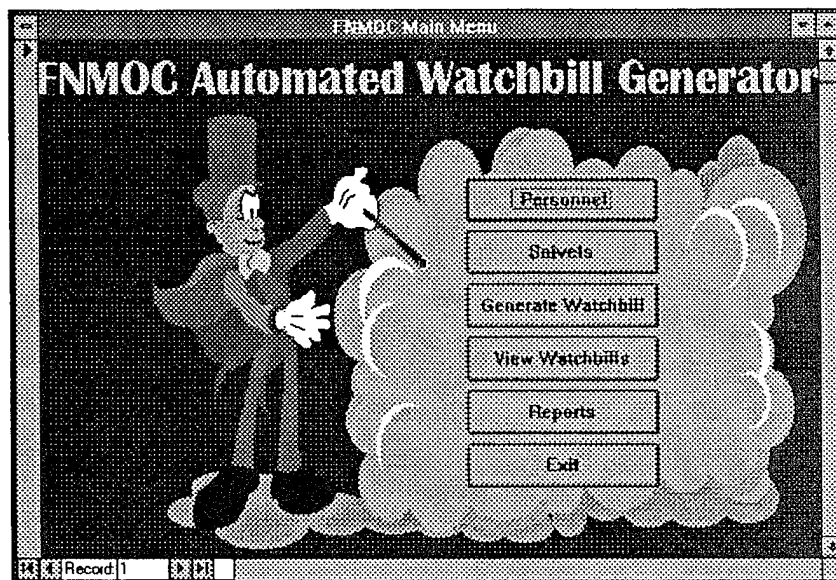
Incorporated into the FNMOCAUTWATGEN is the "Snivel" option. A "Snivel" is a period of time during which a person is unavailable to stand watch. A watchstander may submit a "Snivel" for a variety of reasons: TAD, leave, sickness or family emergency. Before a watchstander is assigned to a watch, his or her "Snivel" record is checked to make sure the watchstander is available to stand the specific watch.

The FNMOCAUTWATGEN also checks to see if the watchstander has been assigned a watch on that day, the previous, or following days. Thus, the dilemma of "back-to-back" watches is eliminated.

The FNMOCAUTWATGEN takes the hassle out of creating watchbills. Simply follow the directions on the user-friendly screens and the system will produce an unbiased, efficient watchbill.

### C. USING THE FNMOC AUTOMATED WATCHBILL GENERATOR

After opening db2.mdb, you are greeted with the FNMOC Main Menu screen shown below.

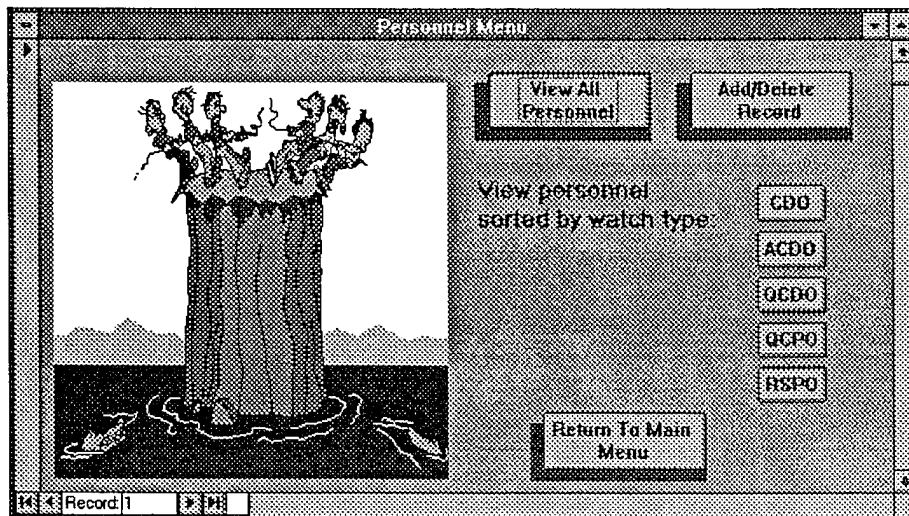


**Main Menu**

Data in the FNMOCAUTWATGEN is organized in four categories: Personnel, Snivels, Watchbills and Reports. The Personnel category consists of all data relating specifically to the watchstanders such as Name, SSN, PRD, etc; snivels are dates when a watchstander is unavailable to stand watch; watchbills are the assignment of watchstanders to the specific watches; and the reports present the data as meaningful information that can be used and distributed. Choose the category which contains the information you wish to view or edit.

### 1. Personnel

Selection of the "Personnel" option displays the following Personnel Menu:



**Personnel Menu**

You have the choice of viewing all personnel currently onboard, adding or deleting a watchstander from the database, viewing personnel sorted by watch type, or returning to the main menu.

#### a. View Personnel Onboard

If you wish to view all personnel currently onboard, push "View All Personnel" and the following form will be displayed:



Personnel

View Report Print Report Return To Personnel Menu

Watch Name	Last Name	First Name	SSN	ReptDate	PRD	PtsLastMonth	PtsThisMonth
ACDO	Buchanan	Todd	376-95-5606	30-Jan-95	23-Aug-97	12	6
ACDO	Dashnerwert	Joe	655-96-0077	23-Feb-94	10-Mar-96	12	6
ACDO	Dimitrou	Dominique	225-35-8745	01-Apr-95	30-Apr-98	12.5	6
ACDO	Garnet	Dan	199-56-4495	01-Mar-94	01-Mar-97	12.5	6
ACDO	Kuypers	Mark	227-45-9706	30-Jul-95	30-Jul-98	12.5	6
ACDO	Loeffler	Susan	276-84-0087	31-Jan-95	31-Jan-98	12	6
ACDO	McDowell	Scott	436-22-2975	01-Dec-94	30-Nov-97	12	6
ACDO	Mechels	Paul	764-29-7539	14-Sep-94	01-Sep-97	15	6.5
ACDO	Seibel	Scott	328-64-8834	22-May-95	01-Jun-96	15	6.5
ACDO	Szczepanick	Stephanie	877-35-1952	01-Feb-95	01-Feb-96	12.5	7.5
ACDO	Uiles	Barry	664-97-6731	24-Oct-95	31-Oct-98	13	6.5

Record 1 of 11

Personnel Form

The form is sorted by type of watchstander and allows you to scroll through the records using the side scroll bars.

### b. Adding or Deleting Personnel

If you wish to add a new watchstander to the database, or remove a watchstander, choose "Add/Delete Record," and the following Add/Delete form will appear.

Personnel

Personnel

Last Name: Whitenham

First Name: Lory

SSN: 225-97-6543

Rank/Rate: 03

Watch Name: CDO

Rept Date: 20-Mar-95

PRD: 01-Oct-97

Pts Last Month: 9.5

Pts This Month: 9

Add New Record Delete Current Record

Return To Personnel Menu

Record 1 of 11

Add/Delete Form

To add a watchstander, choose "Add New Record" and a blank form will appear in which you can add the required information. To delete a watchstander, simply scroll to the record you wish to delete and push "Delete Current Record." Changes to the database are automatically saved when you push any button on the screen.

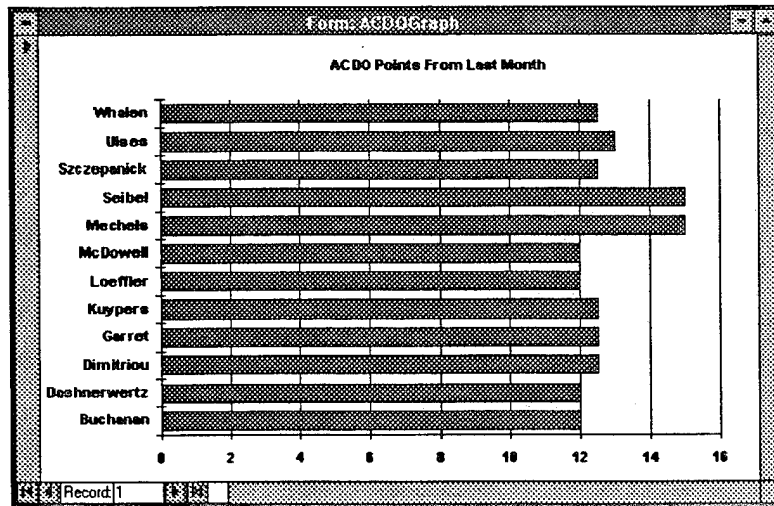
### c. View Records Sorted By Watch Type

If you wish to view personnel sorted by watch type, choose the appropriate category and a form similar to the one shown below will be displayed.

ACDOs				
ACDOs				
Last Name	First Name	DOB	Post with unit	Post Standards
McDowell	Scott	496-22-2975	12	8
Loeffler	Susan	276-84-0087	12	8
Dashnerwell	Joe	655-98-0077	12	8
Buchanan	Todd	376-95-5606	12	8
Dimitriou	Dominique	225-35-8745	12.5	8
Whalen	Tim	387-32-7691	12.5	8
Garret	Dan	199-56-4485	12.5	8
Kuypers	Mark	227-45-9788	12.5	8
Szczepanick	Stephanie	877-35-1952	12.5	7.5
Uiles	Bary	664-97-6731	13	6.5

ACDO Form

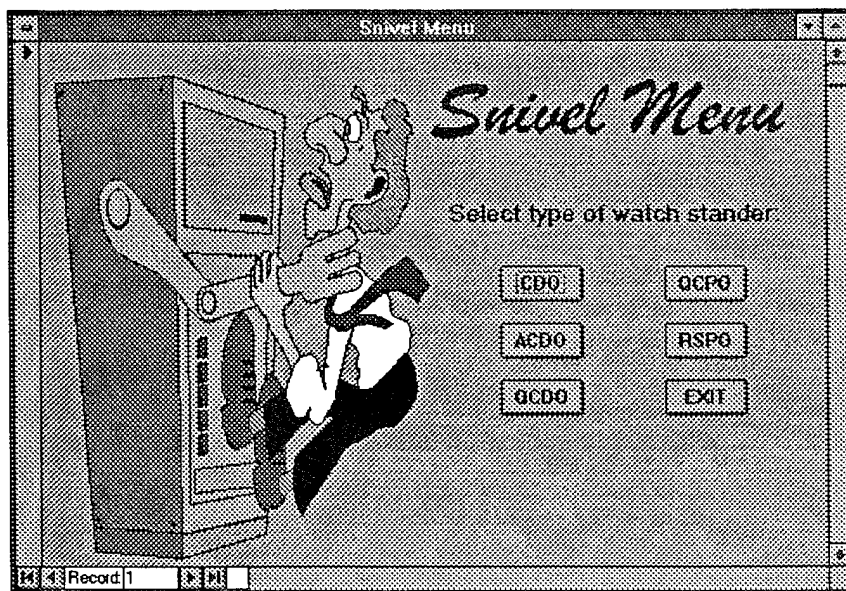
Graphs allow you to show trends in your data or show relationships between different types of data. If you wish to see a graph comparing all watchstanders and their respective point totals from the previous month, push "View Graph," and a graph similar to the one below will be generated.



ACDO Graph

## 2. Snivels

Selection of the "Snivel" option displays the following Snivel Menu:



Snivel Menu

A "Snivel" is a period of time during which a person is unavailable to stand watch. A watchstander may submit a "Snivel" for a variety of reasons: TAD, leave, sickness or family emergency. Select which type of watchstander snivels you wish to edit or view and

a form similar to the CDO Snivel form will be displayed.

LastName:	Personnel ID_FK1:	WatchName:	StartDate:	EndDate:
Eleuterio	342-96-2468	CDO	07-Jan-96	09-Jan-96
Eleuterio	342-96-2468	CDO	20-Feb-96	27-Feb-96
Moore	285-99-7433	CDO	01-Feb-96	03-Feb-96
Mykyta	220-98-7764	CDO	24-Jan-96	28-Jan-96

**CDO Snivels**

To add a snivel, choose "Add Snivel" and a blank form will appear for you to add data. To delete a snivel, simply scroll to the record you wish to delete and push "Delete Snivel." Changes to the database are automatically saved when you return to the Snivel Menu.

### **3. Generating Watchbills**

Selection of the "Generate Watchbill" option displays the following Select Month form.

Form: SelectMonth

1992

Select Month:

☒ Jan.
 ☐ Jul.  
☐ Feb.
 ☐ Aug.  
☐ Mar.
 ☐ Sep.  
☐ Apr.
 ☐ Oct.  
☐ May.
 ☐ Nov.  
☐ Jun.
 ☐ Dec.

Return

Record: 1

**Select Month**

Double click on the month you wish to generate and the Get Value form will be displayed.

Form: GetValueForm

The Automated Watchbill Generator automatically assigns the following default values for specific watches:

Value for weekday watch: 1  
 Value for weeknight watch: 1.5  
 Value for weekend day watch: 2  
 Value for weekend night watch: 25

You may accept default values or modify them as you desire. When finished push Save to continue.

Return To Select Month Menu
 Save Current Values

Record: 1

**Get Value**

FNMOCAUTWATGEN assigns the default values shown above. To edit values click on the value you wish to change and enter the new value. Once all values are correct, push

"Save Current Values" and the Type Of Watch form will be displayed.

Form: TypeOfWatchForm

Select type of watchbill you wish to generate

CDO ACDO QCDO QCPO RSPO EXIT

Record: 1

### Type Of Watch

Select the type of watchbill you wish to create and the Pre-Assign form will be displayed.

Form: PreAssign

If you wish to assign a watchstander to a specific watch before the watchbill is generated, push "Pre Assign." When assignment is completed, push "Generate."

Pre-Assign Generate Return Generate With No Pre-Assign

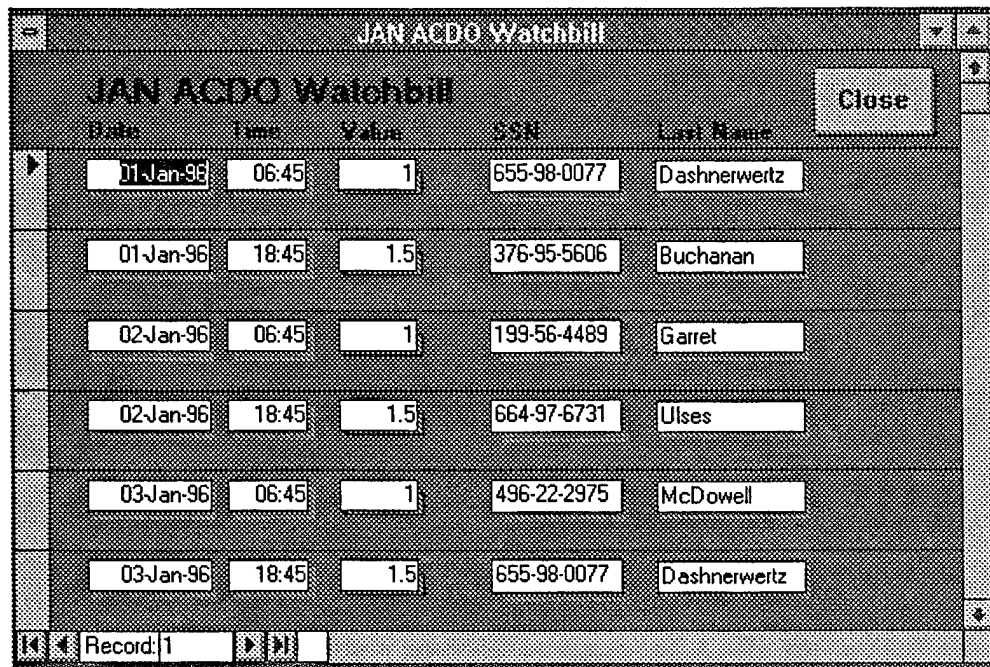
If you wish to generate the watchbill without pre-assigning watches, push "Generate With No Pre-Assign"

Record: 1

### Pre-Assign



If you wish to pre-assign a watchstander to a specific watch, say for example someone volunteers for a holiday, push "Pre-Assign" and a blank watchbill will be displayed. Enter the SSN of the watchstander in the specific watch, push "Close," and you will be back on the Pre-Assign form. Push "Generate" and the program will generate the watchbill and display it in the following format:



Date	Time	Value	SSN	Last Name
01-Jan-96	06:45	1	655-98-0077	Dashnerwertz
01-Jan-96	18:45	1.5	376-95-5606	Buchanan
02-Jan-96	06:45	1	199-56-4489	Garret
02-Jan-96	18:45	1.5	664-97-6731	Ulises
03-Jan-96	06:45	1	496-22-2975	McDowell
03-Jan-96	18:45	1.5	655-98-0077	Dashnerwertz

Record: 1

**JAN ACDO Watchbill**

If you don't wish to pre-assign watchstanders, push "Generate With No Pre-Assign" and the above watchbill will be displayed.

#### 4. View Watchbills

Selection of the "View Watchbills" option displays the following View Watchbills Menu.

Form: ViewWatchbills

**Watchbills**

Select WatchType: OCRO

Select Month:

- ☐ Jan
- ☐ Feb
- ☐ Mar
- ☐ Apr
- ☐ May
- ☐ Jun
- ☐ Jul
- ☐ Aug
- ☐ Sep
- ☐ Oct
- ☐ Nov
- ☐ Dec

Return

Record: 1

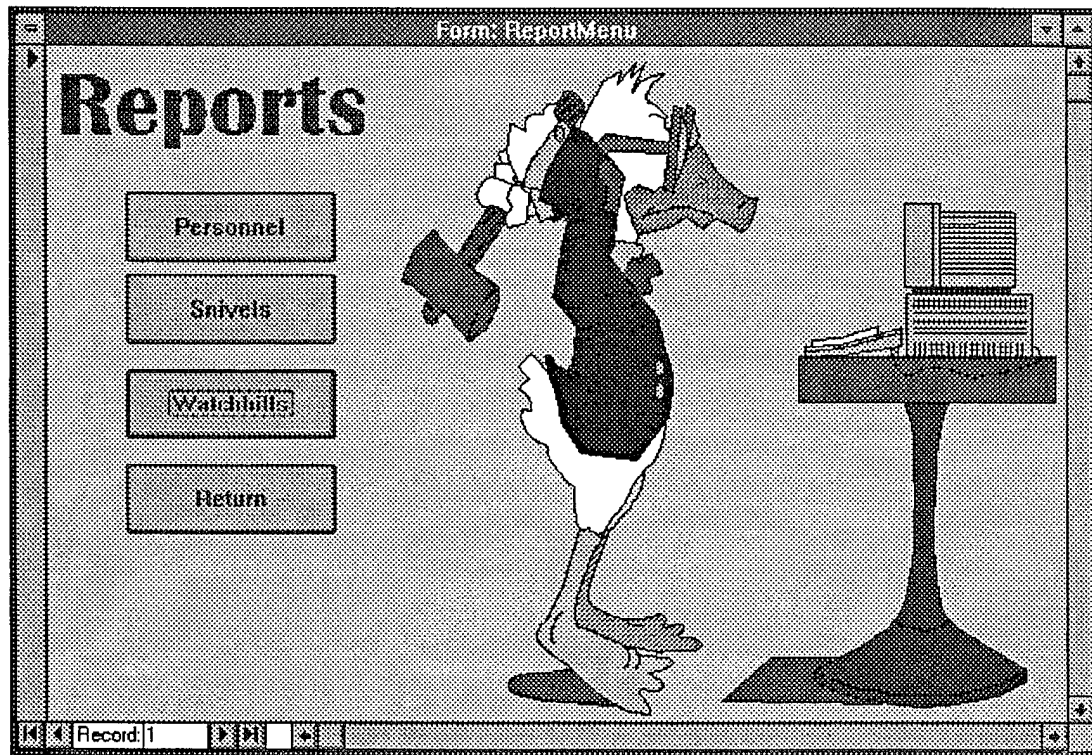
**View Watchbills**

Select the watch type from the pull down menu, and double click on the appropriate month and the appropriate watchbill will be displayed. Keep in mind that the program generates a recommended watchbill. If you wish to modify the watchbill after it has been created, simply delete the Name and SSN the program assigned and type in the data you wish. However, you must then update that watchstander's "PtsThisMth" total via the Personnel Menu.



## 5. Reports

Selection of the "Reports " option displays the following Report Menu.



**Report Menu**

Reports have been designed that present information the way you want. The reports give you a way to retrieve and present data as meaningful information that you can use and distribute.

### **a. Personnel**

If you wish to view or print Personnel Reports, push "Personnel" and the Personnel Reports form will be displayed.

Form: PersonnelReports


# Personnel Reports

Select Type of Report:

Personnel	QCDO
CDO	QCPD
ACDO	RSPD

Return

Record: 1



## Personnel Reports

Double click on the type of personnel report you wish to view and a report similar to the ACDO Report will be displayed.

Report: ACDOReport

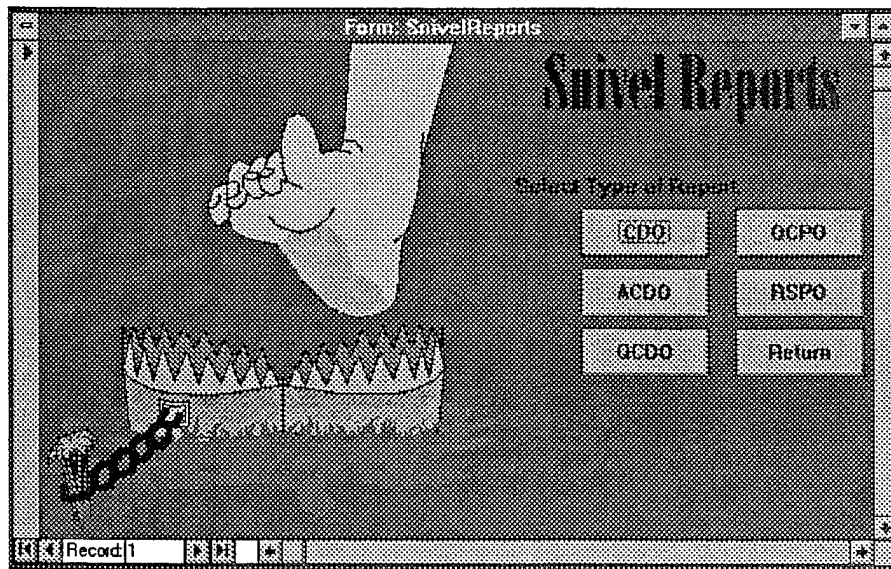
ACDO Report				
15-Aug-95				
LastName	FirstName	SSN	PctLastMonth	PctThisMonth
Buchanan	Todd	376-95-56	12	8
Dashnerwitz	Joe	655-98-00	12	8
Dimitriou	Dominique	225-35-87	12.5	8
Garret	Dan	199-56-44	12.5	8
Kuypers	Merk	227-45-97	12.5	8
Loeffler	Susan	276-84-00	12	8
McDowell	Scott	496-22-29	12	8
Medhels	Paul	764-29-75	15	6.5
Seibel	Scott	328-64-88	15	6.5
Szerepanick	Stephanie	877-35-19	12.5	7.5
Ulsee	Barry	664-97-67	13	6.5
Whalen	Tim	387-32-76	12.5	8
			153.5	91

Page: 1

## ACDO Report

### b.Snivels

If you wish to view or print Snivel Reports, push "Snivel" and the Snivel Reports form will be displayed.



### Snivel Reports

Double click on the type of snivel report you wish to view, and a report similar to the ACDO Snivel Report will be displayed.

LastName	StartDate	EndDate
Buchanan	10-Feb-96	15-Feb-96
Dashnerwert	22-Jan-96	27-Jan-96
Kuypers	02-Jan-96	08-Jan-96
Whisenant	11-Jan-96	17-Jan-96

### ACDO Snivel Report

#### c. Watchbills

If you wish to view or print Watchbill Reports, push "Watchbills" and the Watchbill

Reports form will be displayed.

Form: WatchbillReports

# Watchbill Reports

Select WatchType: 000

Select Month:

<input type="radio"/> Jan	<input type="radio"/> Jul
<input type="radio"/> Feb	<input type="radio"/> Aug
<input type="radio"/> Mar	<input type="radio"/> Sep
<input type="radio"/> Apr	<input type="radio"/> Oct
<input type="radio"/> May	<input type="radio"/> Nov
<input type="radio"/> Jun	<input type="radio"/> Dec

Return

Record: 1

### Watchbill Reports

Select the watch type from the pull down menu, and double click on the appropriate month and a report similar to the Jan CDO Watchbill will be displayed.

Report: JanCDOWatchbill

### Jan CDO Watchbill

15 Aug 95

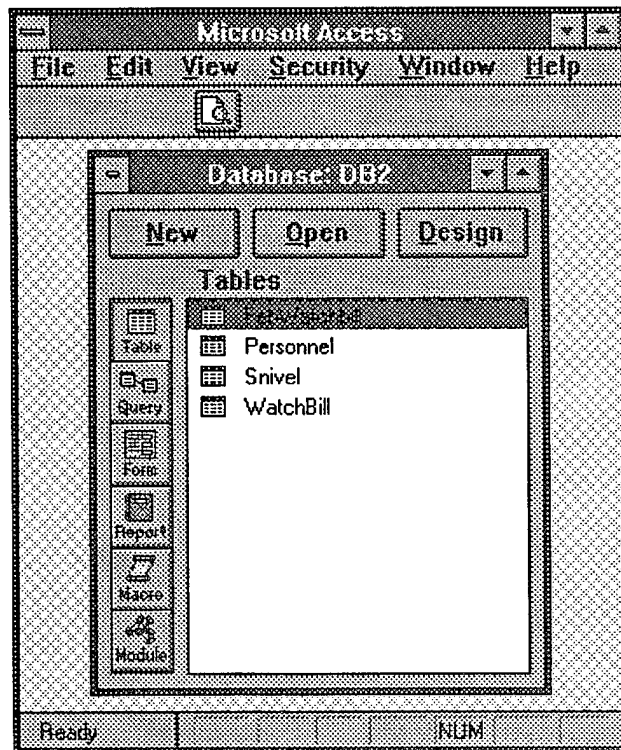
StartDate	StartTime	LastName
01-Jan-96	06:45	Cannon
	18:45	Mays
02-Jan-96	18:45	Smith
	06:45	Eleuterio
03-Jan-96	06:45	Mays
	18:45	Oschmann
04-Jan-96	18:45	Corkhill
	06:45	Moore

Page 1

### Jan CDO Watchbill

## 6. Exit

Selection of the "Exit" option displays the following Database window.



**Database Window**

To close the database, choose Close Database from the File menu.





## LIST OF REFERENCES

- Bow, J. (1966). The heuristic academic master scheduler. Unpublished master's thesis, Naval Postgraduate School, Monterey, CA.
- Dewitz, S. & Olson, M. (1994). Semantic Object Modeling With SALSA. New York: McGraw Hill.
- Elmasri, R & Navathe, B. (1989). Fundamentals of database systems.
- FNOCINST 3120.1. (1993). Fleet Numeric Oceanography Center Standard Operations and Regulations Manual.
- Hartman, J. & Howard, G. (1976). NTDS computer facilities scheduling. Unpublished master's thesis, Naval Postgraduate School, Monterey, CA.
- Hartman, J. & Howard, G. (1977). NTDS computer facilities scheduling final report. Unpublished master's thesis, Naval Postgraduate School, Monterey, CA.
- Huxley, S. & Taylor, P. (1989). A break from tradition for the San Francisco police: patrol officer scheduling using an optimization-based decision support system. Interfaces, 19, 4-24.
- O'Brien, J. (1969). Scheduling handbook. New York: McGraw-Hill.
- OPNAVINST 3120.32. (1994). U. S. Navy Standard Operations and Regulations Manual.
- Poshyanonda, T. (1991). Integration of artificial intelligence and operations research in production planning, scheduling, and control (Doctoral dissertation, University of Missouri, 1989). UMI Dissertation Services.



